

FSA Integration Partner
United States Department of Education
Federal Student Aid



EAI Configuration Management Plan (Production Code Baseline)

Task Order #117

Deliverable # 117.1.4

Version 1.1

April 3, 2003



Table of Contents

1	INTRODUCTION	3
1.1	PURPOSE	3
1.2	SUMMARY	3
2	SOFTWARE CONFIGURATION APPROACH.....	5
2.1	OVERVIEW	5
2.2	HIGH LEVEL SCM PROCESS	6
2.3	SOFTWARE COMPONENTS.....	7
2.4	ROLES & RESPONSIBILITIES.....	7
2.5	SOFTWARE CONFIGURATION MANAGEMENT TOOLS	8
2.6	EAI ENVIRONMENTS	8
3	VERSION CONTROL.....	10
3.1	REPOSITORY STRUCTURE	10
3.2	ACCOUNTS AND PERMISSIONS.....	11
3.3	LABELS AND BRANCHES	11
3.4	CLEARCASE INSTRUCTIONS FOR DEVELOPERS	13
4	PACKAGING	14
4.1	EXTRACTION	14
4.2	COMPILATION.....	14
4.3	EAI SOURCE CODE	14
4.4	MQSERIES OBJECTS	14
4.5	MQSI MESSAGE FLOWS / SETS.....	14
5	MIGRATION.....	15
5.1	MIGRATION REQUESTS	15
5.2	MIGRATION PROCEDURES	17
5.3	MIGRATION CHECKLIST / VALIDATION PROCEDURES	17
6	CHANGE CONTROL	18
6.1	EMERGENCY CHANGE.....	18
6.2	NON-EMERGENCY CHANGE.....	19



APPENDIX A: EAI PRODUCTION CODE CONFIGURATION ITEM INDEX (CII)	21
APPENDIX B: CLEARCASE UNIX COMMAND LINE INTERFACE COMMANDS.....	35
APPENDIX B: CLEARCASE UNIX COMMAND LINE INTERFACE COMMANDS.....	35
APPENDIX C: SOURCE CODE CHECK IN CHECKLIST	36
APPENDIX D: MIGRATION CHECKLIST	37
APPENDIX E: MQSI MIGRATION PROCEDURES.....	38
APPENDIX F: EAI BUILD PACKAGE & MIGRATION PROCEDURES.....	45
APPENDIX G: MIGRATION VALIDATION PROCEDURES	49



1 Introduction

1.1 Purpose

This document's scope focuses on the configuration management of the EAI production code baseline is intended to give an EAI team member, upon their first reading, the basic understanding of what is expected of them if they were required to modify the EAI production baseline.

Within the broader realm of configuration management, there are certainly other documents that must be referenced here and read by all EAI team members including the EAI team's traditional CM plan (for overall CM of EAI "work products" beyond the production baseline). Another document that is referenced extensively is the FSA Enterprise Change Management Process Guide. These other plans remain separate documents that are referenced in this plan where appropriate.

The EAI Team is different from more traditional application development teams in that there is not a clearly defined "beginning and end" for the EAI effort. The EAI Team interfaces with several applications within the FSA portfolio (i.e., FMS, SAIG, COD, etc) and is responsible for a certain subset of production code (the focus of this document) that interconnects, supports and enables (instead of fully comprising) those other systems.

1.2 Summary

The EAI Configuration Management Plan is organized in the following sections:

- **Section 2: Software Configuration Management Approach** – Provides the process and components of EAI Software Configuration Management.
- **Section 3: Version Control** – Provides an overview of the key source code version control components including procedures for checking source code into the version control repository .
- **Section 4: Packaging** – Provides a summary of steps performed to package a build of the EAI architecture.
- **Section 5: Migration** – Provides a summary of the key components involved in a EAI architecture migration and references to detailed migration procedures.
- **Section 6: Change Control** – Provides a high level summary of the processes defined to change the EAI architecture.
- **Appendix A: EAI Production Code Configuration Item Index (CII)**
- **Appendix B: ClearCase UNIX Command Line Interface Commands**



- ***Appendix C: Source Code Check In Checklist***
- ***Appendix D: Migration Checklist***
- ***Appendix E: MQSI Migration Procedures***
- ***Appendix F: EAI Build Package & Migration Procedures***
- ***Appendix G: Migration Validation Procedures***



2 Software Configuration Approach

2.1 Overview

Configuration Management (CM) enables the controlled and repeatable management of information technology (IT) architecture components as they evolve in all stages of development and maintenance. CM is a process by which the project teams and stakeholders identify, communicate, implement, document and manage changes in the systems environment. When properly implemented, CM ensures the integrity of the items that have been placed under its control.

EAI has defined Software Configuration Management (SCM) as the set of CM processes focused on the management of the software components that comprise the EAI architecture. These are the tools, processes, and people involved to ensure the integrity of the EAI architecture as it evolves to support FSA's needs. This document remains focused not on the broad picture of CM at FSA or even the broadest picture of CM at EAI but on the very specific realities of managing the EAI production code baseline (i.e. versioning, packaging and migrating items in the production code baseline). Throughout this document extensive reference to the Teams chosen configuration management tool, Rational ClearCase, are included so that the reader can gain a functional understanding of what is required when utilizing that tool.

Beyond the scope of this document though, it is very important that the reader understand the nature of the “other” major CM documents that this plan references, the broader EAI CM Plan and the Enterprise Change Management Process Guide:

- **Broader EAI CM Plan (for work products beyond the production baseline):**

This traditional plan speaks to the EAI Team’s internal practices for determining and managing “work products” such as the EAI workplan, Task Order, PowerPoint Presentations and all other such files (whereas this plan remains focused on the files that comprise the production baseline). The broader plan contains more detailed and more broadly scoped activities of the EAI Change Control Group as well as broader discussion of the traditional CM practices such as auditing and status reporting of all EAI “work products” etc.

- **FSA Enterprise Change Management Process Guide:**

FSA’s Enterprise-wide change control system is called the Enterprise Change Management (ECM) Tool. The ECM Tool is a web-based interface that allows application teams in the FSA portfolio (such as EAI) to submit formal change requests changes to production code to the Data Center. Indeed the Data Center is responsible for making the actual changes to production code that are requested by Application Team members.

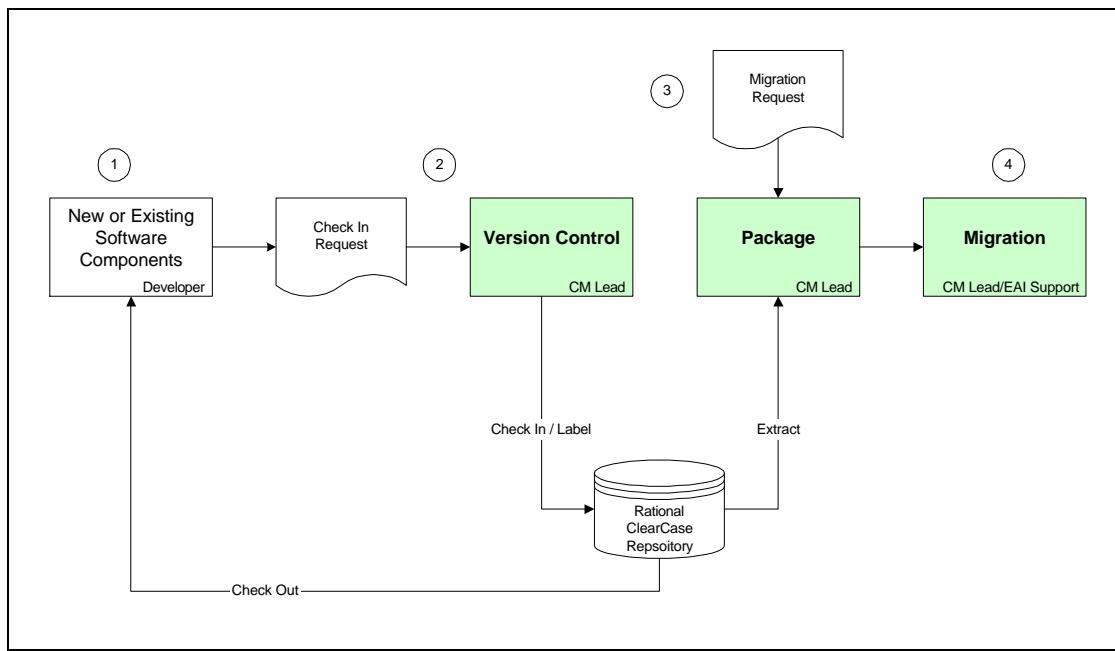
The ECM Tool assigns a unique control number to each request that is made by an application Team of the Data Center. Additionally, the Tool allows the application team



to track the progress of their requested change as the data center takes the necessary steps to implement the change and also provides coordination among other application teams that could be affected by a different team's requested change. The EAI team is the single most prolific user of the ECM Tool among all FSA Application Teams.

2.2 High Level SCM Process

The following diagram outlines the high level SCM process for all software changes to the EAI Architecture. Specifically, the process is facilitated by the EAI team's SCM Tool Rational ClearCase and addresses the key related activities which include versioning, packaging and migrating items in the production code baseline:



1. New or existing software component is created or modified by the EAI Developer. Prior to any modifications on a software component, the EAI Developer will check out the software component in ClearCase to indicate that the EAI Developer is making changes. It is expected that the developer check out the component immediately upon accessing it and check it in immediately upon completion.
2. The Developer submits a Check In request to the CM Lead to have the software component checked into or stored in the SCM Repository (Rational ClearCase). The CM Lead validates the request in a timely manner, ensures all required information is provided, and checks in/labels the software component.
3. A Migration Request initiates the process by which the CM Lead will package the EAI Architecture build and prepare it for migration to the requested environment.



-
4. The CM Lead or other EAI support person will migrate the new build of the EAI Architecture to the requested environment and validate the operability of the newly migrated environment. While the timeframes for expected completion of this activity vary on a case-by-case basis, it is important that these activities are completed in a timely manner.

2.3 Software Components

EAI software components consist of EAI source code, MQSeries object definitions, and MQSeries Integrator (MQSI) message flows. MQSeries object definition scripts are used to create MQSeries queue managers, queues, and channels within MQSeries. MQSI message flow/set scripts are used to create message flows within MQSI. EAI source code consists of the following:

- Java source code
- C++ source code
- UNIX shell (ksh) scripts
- Configuration text files
- XML files
- Prepackaged out-of-the-box helper software

All the EAI software components are managed under CM to control configuration change. The EAI Production Baseline is packaged together to create what is referred to as an EAI build. As outlined below, the EAI CM Lead is responsible for authorizing the “Checkout” of files under SCM. A request is made of the CM Lead by a team member and the CM Lead records the request and the associated activity/changes to the checked-out file in the SCM Configuration Item Index (CII). Only the CM lead is authorized to make updates to the SCM CII.

2.4 Roles & Responsibilities

The following section contains information on roles and responsibilities associated with EAI team SCM of all the software components identified above.

2.4.1 EAI CM Lead:

- Software component version control
- Perform software component check ins as requested
- Package builds for Development, SIT, UAT, IST, and PROD
- Migrate builds to Development, SIT, UAT, and IST environments as requested
- Stage builds for Production migration as requested
- Migration validation

2.4.2 EAI Developers:

- Create and modify software components
- Maintain list of files that contain passwords
- Create list of files for CM Lead to label
- Submit EAI Source Code Check-In Requests to CM Lead via email to check-in/check-out code



-
- Submit Migration Requests to CM Lead via email for EAI build migration

The ClearCase system administrators at the Virtual Data Center (VDC) are Rebecca Panetta(rpanetta@csc.com) and Sandra Sterantino (ssterant@csc.com). The VDC owns and is responsible for all third party vendor software.

2.5 Software Configuration Management Tools

2.5.1 Rational Clearcase

The primary tool used by the EAI team to manage the EAI architecture software components is Rational ClearCase. ClearCase is an off-the-shelf product that provides the key SCM capabilities of version control. This includes tracking of all software development objects, use of permanent, secure data repositories, and support for parallel development. The EAI implementation uses ClearCase hosted on a Sun Solaris UNIX platform. Although a graphical user interface for the tool exists, the EAI team primarily manages the repository via the command line interface (see *Appendix B – ClearCase UNIX Command Line Interface Commands* for commonly used commands). The CM Lead for the EAI Team is responsible for maintaining the integrity of the EAI Rational ClearCase repository.

The CM Lead is also responsible for ensuring the EAI team members have access to the Rational Software (as well as internally developed) training material including detailed instructions on the use of ClearCase.

2.5.2 MQSoftware QPasa!

MQSoftware QPasa! is a new tool currently being implemented that will provide for the version control and deployment of MQSeries objects. QPasa! is a Windows-based tool with agents that run on the various MQSeries supported platforms. EAI will use QPasa! to manage MQSeries object definitions once implemented and integrated into the current EAI SCM processes.

MQSeries object definitions are simple scripts that are created by exporting the object definition of an MQSeries object. These definitions are currently stored within the ClearCase data repository. QPasa! will provide enhanced object definition scripts extraction and migration to the EAI environments once implemented. The CM Lead for the EAI Team will be responsible for managing the MQSoftware QPasa! Tool.

2.6 EAI Environments

The EAI architecture enables systems on varied server platforms to communicate with each other through MQSeries. The EAI architecture is comprised of HP UNIX servers, Sun Solaris UNIX servers, OS/390 IBM mainframes, OpenVMS mainframes, and NT servers.

MQSeries objects are installed on each of these platforms, whereas the EAI source code is only installed on HP UNIX and Sun Solaris UNIX servers. CommerceQuest Data Integrator is a partner product to MQSeries than provides bulk file transfer capabilities over an MQSeries infrastructure. This is a key function of the EAI architecture and is the mechanism used for FSA system to FSA system file transfer and so is installed on all servers. MQSI coupled with EAI source code are used to translate and transform messages.



There are five (5) environments that EAI supports for MQSeries objects and EAI source code each consisting of numerous servers. MQSeries objects and EAI source code must be managed across each of the environments to ensure the integrity of the baseline. The five (5) environments are listed below (The diagram in Section 6.2 displays the environment flow):

- Development (DEV)
- System Integration Test (SIT) – testing environment using stubs
- User Acceptance Testing (UAT) – testing environment
- Inter System Testing (IST) – pre-production environment
- Production (PROD)

2.6.1 MQSeries Queue Managers

MQSeries Queue Managers own and manage MQSeries objects. Each machine can have multiple Queue Managers. The table below illustrates the various environments per system.

Server \ QM	Environment				
System	DEV	SIT	UAT	IST	PROD
EAI 1	SU35E16 \ EAID1	SU35E16 \ EAIS1	SU35E16 \ EAIU1	SU35E16 \ EAII1	SU35E14I \ EAIP1
EAI 2	SU35E17 \ EAID2	SU35E17 \ EAIS2	SU35E17 \ EAIU2	SU35E17 \ EAII2	SU35E3 \ EAIP2
SAIG	HPL-15 \ SAIGD1	HPL-15 \ SAIGS1	HPL-15 \ SAIGU1	HPL-15 \ SAIGI1	HPL-16 \ SAIGP2
PEPS	HPK1 \ PEPSD1	HPK1 \ PEPSS1	HPK1 \ PEPSU1	HPK1 \ PEPSI1	HPK2 \ PEPSP1
CPS	CPSED \ CPD1	N/A	CPSED \ CPA1	CPSED \ CPT1	CPSED \ CPP1
NSLDS	NSLSYSP \ NPD1	N/A	NSLSYSP \ NPA1	N/A	NSLSYSP \ NPP1
COD	VD0Q	N/A	UCSV	UCSV	PD0Q
LOWeb	HPL12 \ LOWD1	HPL12 \ LOWS1	HPL12 \ LOWU1	HPL12 \ LOWI1	HPL13 \ LOWP1
FMS	HPL6 \ FMSD1	HPL6 \ FMSS1	HPL6 \ FMSU1	HPL6 \ FMSI1	HP-V2 \ FMSP1
eCB 1	SU35E5 \ ECBSD1	SU35E5 \ ECBSS1	SU35E5 \ ECBSU1	SU35E5 \ ECBSI1	SU35E9 \ ECBSP1
eCB 2	SU35E5 \ ECBSD2	SU35E5 \ ECBSS2	SU35E5 \ ECBSU2	SU35E5 \ ECBSI2	SU35E13 \ ECBSP2
DLSS	CRDEV2 \ DLSST1	N/A	N/A	N/A	CRPAS1 \ DLSSP1
WAS 1	SU35E5 \ WASD1	SU35E5 \ WASS1	SU35E5 \ WASU1	SU35E5 \ WASI1	SU35E9 \ WASP1
WAS 2	SU35E5 \ WASD2	SU35E5 \ WASS2	SU35E5 \ WASU2	SU35E5 \ WASI2	SU35E13 \ WASP2
Informatica	SU35E5 \ INFD1	SU35E5 \ INFS1	SU35E5 \ INFU1	SU35E5 \ INFI1	SU35E18 \ INFP1

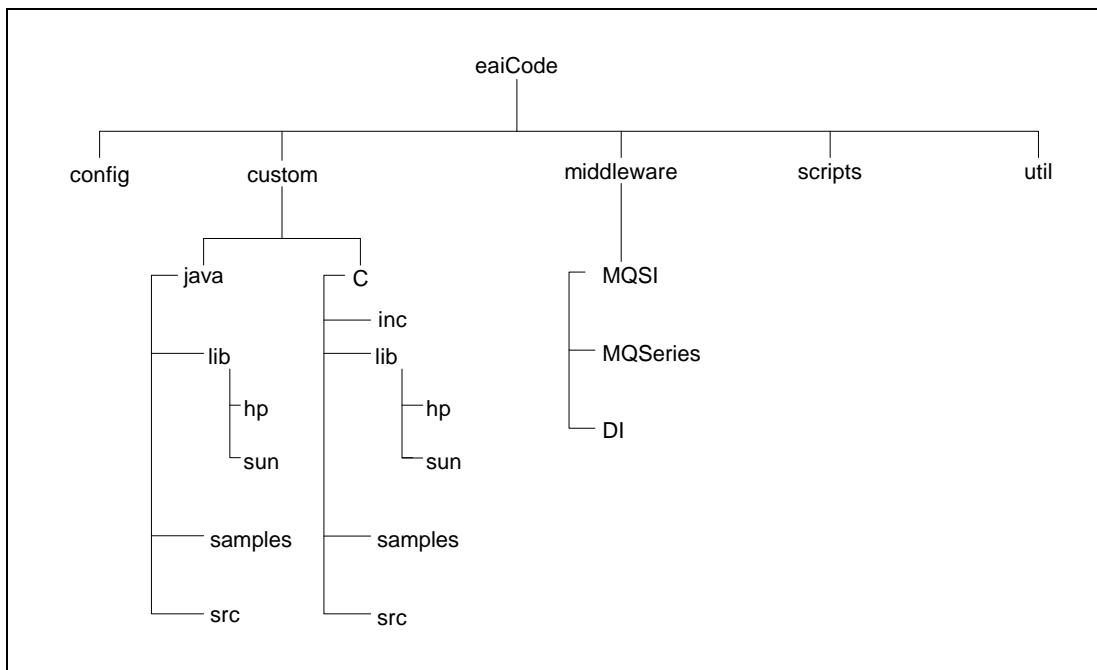


3 Version Control

Rational ClearCase is a software configuration management tool that provides the version control capabilities for the EAI team. Rigorous management of the software components as they change and the ability to create fully operational EAI builds is the heart of version control.

3.1 Repository Structure

The ClearCase repository is where the software components are stored. The repository structure is logically organized to align with the types of source code files stored within it. The large number and types of EAI source code components compared to the MQSeries object definition and MQSI message flow scripts require a more detailed structure. This is represented by the config, custom, and scripts directories. The source code is stored in multiple directories within the ClearCase data repository, which is shown in the diagram below:



- **config:** configuration text files
- **custom:** source code for custom adapters and other custom code
- **middleware:** text file scripts for each middleware application
- **scripts:** UNIX shell scripts
- **util:** prepackaged out-of-the-box utilities and helper software



3.2 Accounts and Permissions

Rational ClearCase and the underlying EAI source code repository run on SUN Solaris UNIX servers SU35E16 and SU35E17. ClearCase access correlates directly with UNIX accounts and permissions. The ClearCase code repository for EAI is owned by the “eaivob” UNIX account with “mqm” as the group. UNIX accounts with “mqm” as their primary group will be able to access the ClearCase code repository for EAI.

3.3 Labels and Branches

3.3.1 Labels

Labels are assigned to each revision of a software component or Configuration item (CI) version controlled in ClearCase. Collectively all CIs with the same label identify a build of the EAI architecture. Each distinct build represents the EAI architecture at a point in time and can be reproduced based on the applied label.

Labeling:

- Increases confidence that the code deployed to an environment is the same code tested in the previous environment.
- Allows to quickly re-build an environment in the event that it is lost or damaged.
- Facilitates branching.

Code is labeled by the CM lead every time a build is ready for migration using the following naming convention:

R<release number>.<fix number><environment>

Label examples:

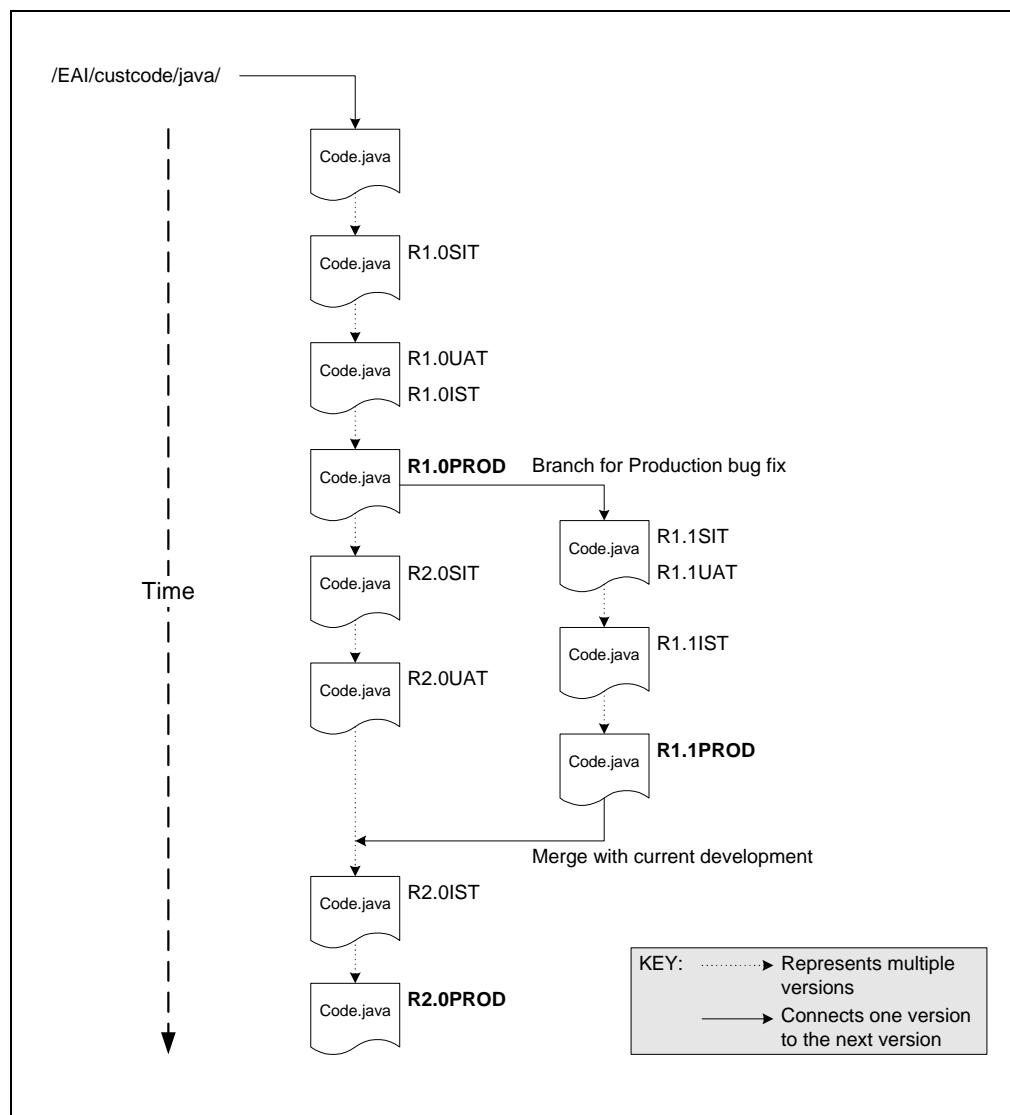
Environment	Label
Release 1 system integration test (a.k.a. assembly test)	R1.0SIT
Release 1 user acceptance test	R1.0UAT
Release 1 inter-system test	R1.0IST
Release 1 production	R1.0PROD
Release 1, 1 st production fix, system integration test	R1.1SIT
Release 1, 1 st production fix, user acceptance test	R1.1UAT
Release 1, 1 st production fix, inter-system test	R1.1IST
Release 1, 1 st production fix, production	R1.1PROD
Release 1, 2 nd production fix, system integration test	R1.2SIT
Release 1, 2 nd production fix, user acceptance test	R1.2UAT
Release 1, 2 nd production fix, inter-system test	R1.2IST
Release 1, 2 nd production fix, production	R1.2PROD
Release 2 system integration test	R2.0SIT



3.3.2 Branches

Code branches will be used to make production bug fixes based on the last production release while continuing development for a next release. We assume that each EAI release may include functionality to meet the interface requirements of multiple FSA initiatives; therefore branches will not be used for parallel EAI development efforts.

The following diagram is an example of how label is applied to a CI as it progresses through the development lifecycle:





3.4 ClearCase Instructions for Developers

3.4.1 Source Code Check In

Once a software component is successfully developed and ready to be version controlled, the EAI developer will submit an EAI Source Code Check-In Request to the CM Lead. One email may be submitted for a check-in request for a set of related files. The detailed steps of the source code check in process are defined in the following table:

Step #	Step	Responsible
1	<p>Create an email request containing the following information:</p> <ul style="list-style-type: none">▪ File Name/s;▪ Reason for Change: (ECM#, Issue, etc)▪ Detailed Description of Change: (Detail the enhancement, Production Bug Fix)▪ Additional Instructions (Optional)▪ Attached file/s <p>The request should be sent</p> <p>To:</p> <ul style="list-style-type: none">▪ Bon Ku <p>CC:</p> <ul style="list-style-type: none">▪ CM Lead Backup (Jeff Goldhirsch)▪ Patrick Volpe▪ Eric Suzuki <p>Subject: ACTION REQUIRED: Source Code Check In – {file name}</p>	EAI Developer
2	Perform Code Check-In Checklist (<i>Appendix C: Source Code Check In Checklist</i>). Contact Requestor or Developer to resolve incomplete checklist items.	EAI CM Lead (Bon Ku)
3	Check source code into ClearCase and label appropriately	EAI CM Lead (Bon Ku)
4	Send email (Reply to All) response to original Check-In Request with notification of check-in completion.	EAI CM Lead (Bon Ku)



4 Packaging

The packaging of an EAI build involves preparing the version controlled software components for migration. Packaging of an EAI build is performed by the CM Lead and is initiated by the submission of a Migration Request. Packaging consists of source code extraction from the ClearCase repository and compilation.

4.1 Extraction

The first step in packaging an EAI Build is extraction. By modifying parameters in ClearCase's configuration specification, copies of the source code can be extracted out of the ClearCase repository based on the previously assigned label or view. After the code has been extracted, it is then copied to a new directory or tar-ed up and ftp-ed to the destination UNIX server designated in the Migration Request. Refer to [Appendix F: EAI Build Package & Migration Procedures, Steps 1-6](#) for detailed steps.

4.2 Compilation

Makefiles provide the instruction for compiling the EAI source code and creating the associated executables. A makefile is used with the UNIX `make` utility to determine which portions of a program to compile. A makefile is basically a script that guides the make utility to choose the appropriate program files that are to be compiled and linked together. The main Makefile for the build is located under the base directory, which is normally defined by the EAIDIR environment variable. The main Makefile will use the `make.inc` files in the \$EAIDIR/scripts directory to execute the other Makefiles in the \$EAIDIR/src directory. After executing the `make`, parameters in specific files need to be modified according to the environment. Refer to [Appendix F: EAI Build Package & Build Procedures, Steps 7-13](#).

4.3 EAI Source Code

EAI source code is packaged by use of the Makefile, which will `tar` the source code and directory structure in a package, which will then be migrated to the destination environment.

4.4 MQSeries Objects

MQSeries object definitions are packaged using a C++ executable (\$EAIDIR/saveqmgr) which extracts object definitions into a text file. MQSeries object definition files are extracted and applied to the affected Queue Manager.

4.5 MQSI Message Flows / Sets

MQSI message flows and sets are packaged using MQSI which extracts message flows and sets into a zip file. MQSI message flows and sets are extracted from ClearCase and ftp-ed to the appropriate Windows NT MQSI configuration control server for import and deployment. Refer to [Appendix E: MQSI Migration Procedures](#)



5 Migration

Migration is the process of moving, installing, and validating a packaged EAI Build to an environment. A release note will be included in each packaged EAI Build. The contents of a release note are listed in [Appendix D: Migration Checklist](#). A migration is initiated by the submission of a Migration Request to the CM Lead. In all cases the requests will be completed by an EAI Developer or member of the EAI team.

5.1 Migration Requests

5.1.1 Migration Request to Development or Test Environment

Step #	Step	Responsible
1	<p>Create an email request containing the following information:</p> <ul style="list-style-type: none">▪ Environment: (Dev, SIT, IST, UAT)▪ Date/Time Need: (mm/dd/yy hh:MM)▪ Server: (i.e., SU35E17)▪ Reason for Migration: (Detail the enhancement, Production Bug Fix, ECM#,etc)▪ Teams/Systems Affected:▪ Dependencies – i.e. MQ object push to X or ECM # <p>The request should be sent</p> <p>To:</p> <ul style="list-style-type: none">▪ Bon Ku <p>CC:</p> <ul style="list-style-type: none">▪ CM Lead Backup (Jeff Goldhirsch)▪ Eric Suzuki▪ Patrick Volpe <p>Subject: ACTION REQUIRED: DEV/TEST MIGRATION REQUEST For {ENVT} On {MM/DD/YY}</p>	Requestor
2	Review the Migration Request for completeness. Return to Requestor for additional detail, if necessary.	EAI CM Lead (Bon Ku)
3	Perform Migration Checklist. Contact Requestor or Developer to resolve incomplete checklist items.	EAI CM Lead (Bon Ku)
4	Label, Package, Create Release Notes, and Document Build for Migration	EAI CM Lead (Bon Ku)
5	Send email Migration Notification to EAI Team or Other Affected Teams.	EAI CM Lead (Bon Ku)
6	If no migration conflicts exist, Migrate EAI Build to requested environment.	EAI CM Lead (Bon Ku)
7	Validate EAI Build migration (Appendix G: Migration Validation Procedures)	EAI CM Lead (Bon Ku)
8	Send email (Reply to All) response to original Migration Request with notification of successful migration completion and environment availability.	EAI CM Lead (Bon Ku)



5.1.2 Migration Request to Production Environment

Step #	Step	Responsible
1	<p>Create an email request containing the following information:</p> <ul style="list-style-type: none">▪ Environment: (PROD)▪ Date/Time Need: (mm/dd/yy hh:MM)▪ Server: (i.e., SU35E3)▪ Reason for Migration: (Detail the enhancement, Production Bug Fix, ECM#,etc)▪ Teams Affected:▪ Dependencies – i.e. MQ object push to X or ECM # <p>The request should be sent</p> <p>To:</p> <ul style="list-style-type: none">▪ Eric Suzuki <p>CC:</p> <ul style="list-style-type: none">▪ Bon Ku▪ Patrick Volpe <p>Subject: ACTION REQUIRED: PRODUCTION MIGRATION REQUEST</p>	Requestor
2	Review the Migration Request for completeness. Return to Requestor for additional detail, if necessary.	EAI Development Lead (Eric Suzuki)
3	Approve Migration Request. Send email (Reply to All) response to original Production Migration Request with Approval or Disapproval/Reason.	EAI Development Lead (Eric Suzuki)
4	Create ECM Request	Requestor
5	Optional: Update ICD, IID, or other affected design document.	EAI Developer
6	Perform Migration Checklist. Contact Requestor or Developer to resolve incomplete checklist items.	EAI CM Lead (Bon Ku)
7	Label, Package, Create Release Notes, and Document Build for Migration	EAI CM Lead (Bon Ku)
8	Send email Migration Notification to EAI Team or Other Affected Teams.	EAI CM Lead (Bon Ku)
9	Once ECM is approved and scheduled, create EAI Build to requested environment.	EAI CM Lead (Bon Ku)
10	Validate EAI Build migration (<u>Appendix G: Migration Validation Procedures</u>) (also see Section 5.3 Migration validation/checklist procedures)	EAI Primary/Secondary Support
11	Update ECM to Closed	Requestor



5.2 Migration Procedures

5.2.1 EAI Source Code

Migrating EAI source code to an environment involves the following high level steps:

1. Ftp the packaged code (tar file) to the destination system.
2. Stop any running processes for the code being deployed.
3. Make a backup of the existing code.
4. Extract (untar) the packaged code in the appropriate directory of the destination system.
5. Start any processes that should be running.

Refer to [Appendix F: EAI Build Package & Migration Procedures](#) for the detailed procedures.

5.2.2 MQSeries Objects

MQSeries Object definition files are applied to the using the MQSeries `rwmqsc` facility.
Refer to the *MQSeries MQSC Command Reference, Running an MQSC command file*.

5.2.3 MQSI Message Flows / Sets

MQSI Message flow and sets are applied to enable or change existing message transformation functionality. Refer to [Appendix E: MQSI Migration Procedures](#) for detailed procedures for migration.

5.3 Migration Checklist / Validation Procedures

The CM lead is responsible for performing the Migration Checklist and Migration Validation Procedures. The checklist and validation procedures help to ensure the repeatability and consistency each migration. Validation is performed upon successful migration of an EAI Build into an environment to ensure EAI operability.

Refer to [Appendix D: Migration Checklist](#)

Refer to [Appendix G: Migration Validation Procedures](#)



6 Change Control

Changes to the EAI architecture are classified as either emergency or non-emergency (see Section 6.1 and 6.2). Implementing a change uses all the major elements of the SCM, version control, packaging, and migration. Once a Production issue has been identified, the EAI team will perform a root cause analysis and make the recommended fix. Depending on the impact to the FSA, the severity of the issue will be assigned as either emergency or non-emergency. The governing guidance with regard to determination of emergency vs. non-emergency changes are referenced in the Sections 6.1 and 6.2.

With regard to broader issues of change management, as stated previously FSA's ECM Tool is the tool that facilitates a change request's submission, review by other affected parties across the enterprise and ultimate authorization/approval for the change to be implemented. The data center determines different standards for development vs production change requests and the requirements that need to be met to move code from one environment to another.

From the perspective of the EAI Team, the process involves:

1. Submission of the CR into the ECM Tool (see the final step shown in the “non-Emergency” process flow in Section 6.2)
2. Review of the submitted CR by Data Center resources
3. “Proposal of the CR (via automatically generated ECM Tool notification emails) to the other enterprise-wide stakeholders who may be affected by the “proposed” change
4. Approval of the CR by the Data Center (upon determination that other applications/stakeholders are not adversely affected)
5. Implementation of the requested change by the data center
6. Notification of the application Team (via automatically generated ECM Tool notification emails) that the change has been implemented by the data center
7. Validation by the EAI Team member who originally submitted the request that the change was implemented successfully by the data center

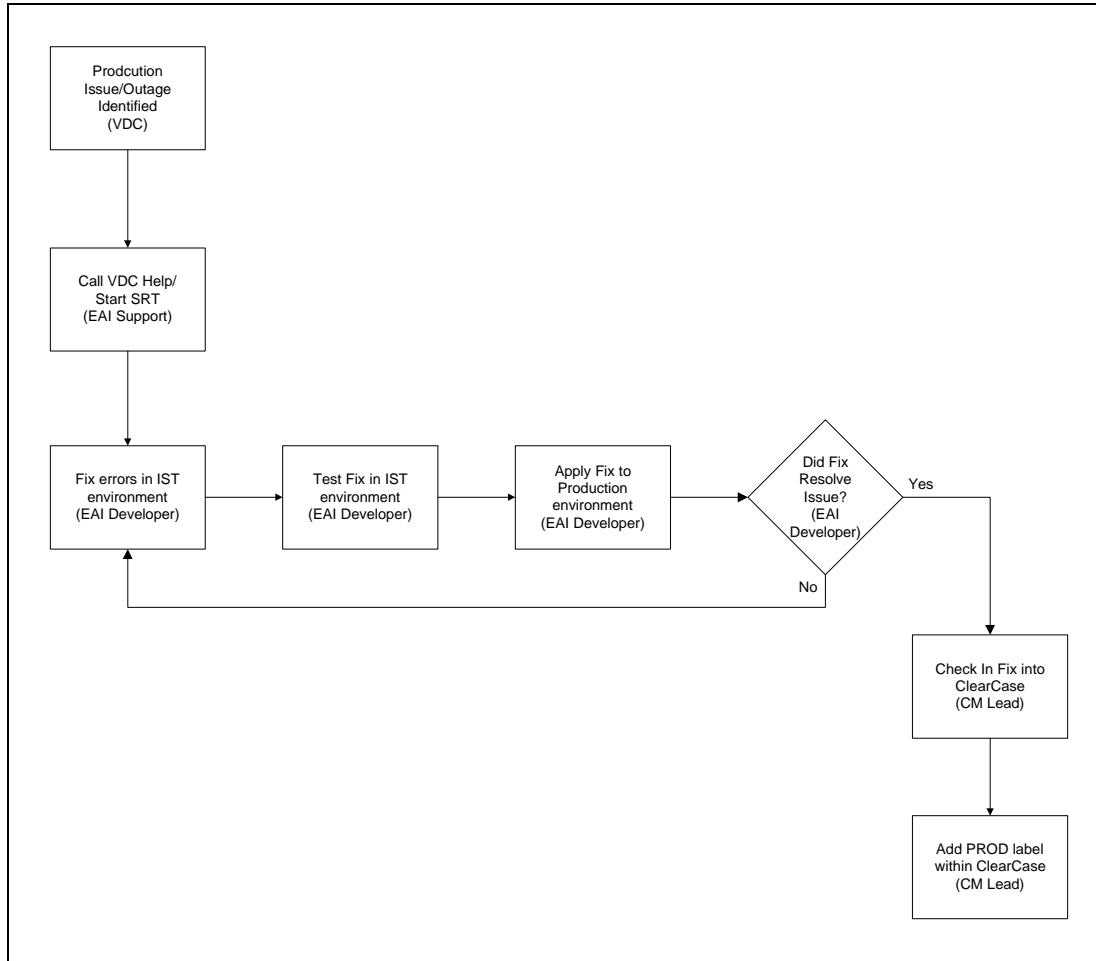
6.1 Emergency Change

The term “Emergency Change” (as well as normal change – see below) was established and is defined by CSC, the contractor that runs the VDC. As CSC changes these definitions, this plan will be updated. For additional information about CSC’s latest definition of these terms, please contact Gary Adams of CSC at gadams2@csc.com. Presently, CSC’s expectations are that they be given approximately 14 days notice/lead time for normal/non-emergency changes (see Section 6.1) but no such lead time is required for emergency changes.

Emergency fixes assume that an outage in the Production environment and immediate action is required to restore normal operations. Emergency changes to the EAI architecture follow the VDC defined System Restoration Team (SRT) process (for additional info, contact Gary Adams). The diagram below illustrates the process flow for non-emergency changes in Production.



The primary characteristic of an emergency change is that Emergency Changes are distinguished from non-emergency changes in that “emergencies” are immediately “called-in” to the Data Center’s command Center/Help Desk for immediate resolution (as shown in diagram). The individual responsible for each action is shown in parenthesis under each action.

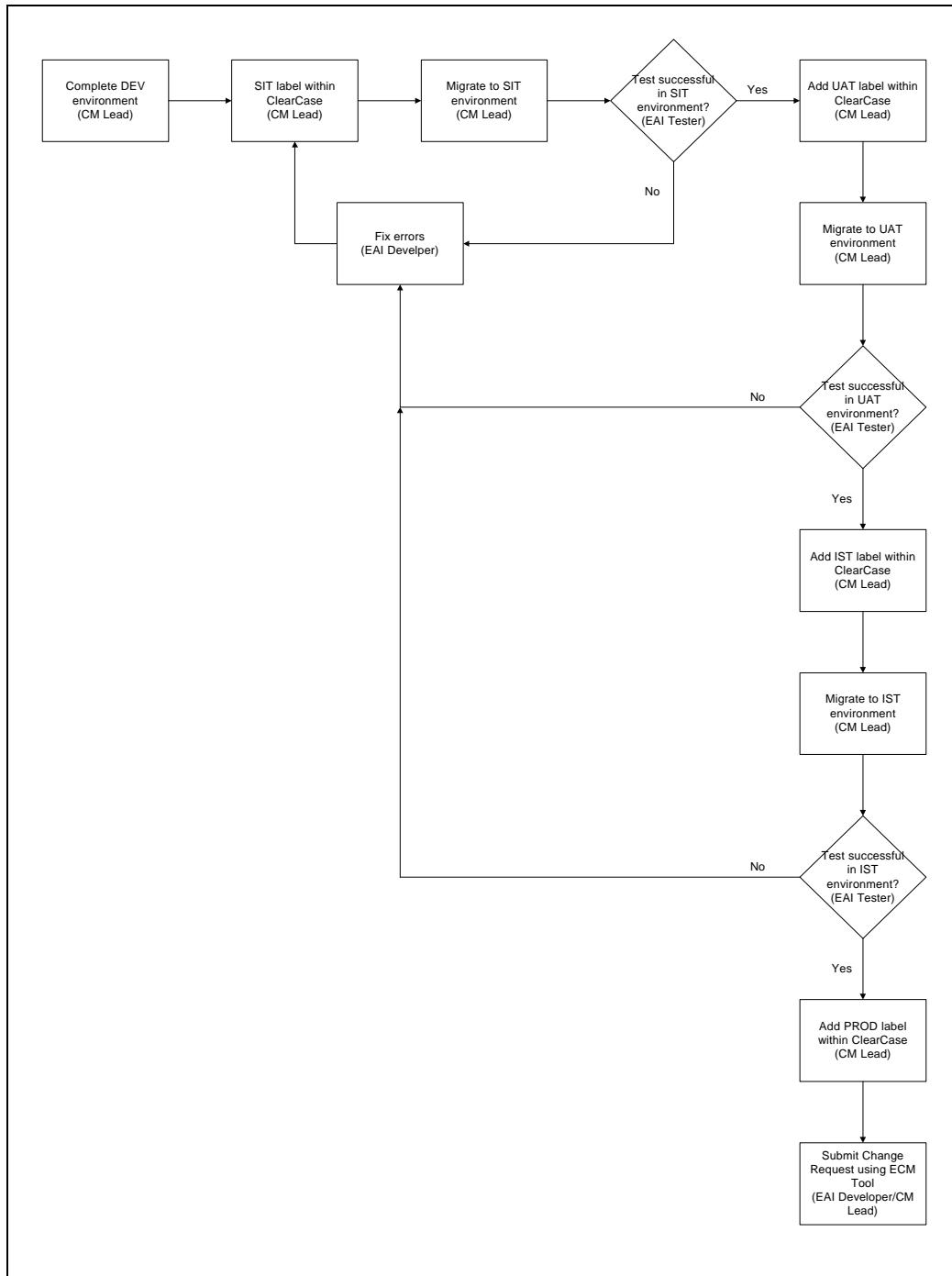


6.2 Non-Emergency Change

Non-emergency fixes are treated as normal changes to the EAI architecture and follow the defined Virtual Data Center process for “Normal” changes. The diagram below illustrates the process flow for non-emergency changes in Production.

The FSA Enterprise Change Management (ECM) process involves the entry of the CR into the ECM Tool and allows the submitter to identify the “Priority” of the CR (Normal vs. Emergency). The individual responsible for each action is shown in parenthesis under each action.

The diagram begins with the step of the development environment being identified as complete.





APPENDIX A: EAI Production Code Configuration Item Index (CII)

The following table provides the list of production code files over which the EAI Team maintains configuration management. Changes to these files are tracked and reported on by the CM Lead. Periodic Audits are performed in accordance with the EAI team's broader CM planning and procedures prior to each major release. When a new baseline is taken, the CM Lead should update the EAI Production CII and current date of the new baseline. The following files comprise the EAI production Code/SCM baseline as of 4/1/03.

Control #	Filename	Revision	Location	Date	Changes
1	.	9	su35e16/su35e17	4/1/2003	
2	admin	1	su35e16/su35e17	4/1/2003	
3	scripts	4	su35e16/su35e17	4/1/2003	
4	eaiCodeDeploy.sh	5	su35e16/su35e17	4/1/2003	
5	eaiEnginePropConfig.sh	1	su35e16/su35e17	4/1/2003	
6	eaiSAGjobConfig.sh	9	su35e16/su35e17	4/1/2003	
7	eaiSiteSpecific.sh	3	su35e16/su35e17	4/1/2003	
8	switchEnv.sh	1	su35e16/su35e17	4/1/2003	
9	eaiCode	1	su35e16/su35e17	4/1/2003	
10	Makefile	20	su35e16/su35e17	4/1/2003	
11	README	1	su35e16/su35e17	4/1/2003	
12	config	11	su35e16/su35e17	4/1/2003	
13	EAICONF.INI	3	su35e16/su35e17	4/1/2003	
14	PEPSDELTA.properties	1	su35e16/su35e17	4/1/2003	
15	PEPSSHIELD.properties	1	su35e16/su35e17	4/1/2003	
16	PEPSSHIELDFULLREFRESH.properties	1	su35e16/su35e17	4/1/2003	
17	SAIGftfconfig.ini	1	su35e16/su35e17	4/1/2003	
18	amt.dtd	1	su35e16/su35e17	4/1/2003	
19	amtcod.xml	2	su35e16/su35e17	4/1/2003	
20	amtcod_rel20.xml	1	su35e16/su35e17	4/1/2003	
21	amtcod_rel21.xml	1	su35e16/su35e17	4/1/2003	
22	amthost.xml	2	su35e16/su35e17	4/1/2003	
23	dirmon.ini	1	su35e16/su35e17	4/1/2003	
24	eaiami.properties	2	su35e16/su35e17	4/1/2003	
25	engine.properties	12	su35e16/su35e17	4/1/2003	
26	fms.properties	3	su35e16/su35e17	4/1/2003	
27	ftfconfig.ini	4	su35e16/su35e17	4/1/2003	
28	mqipool.loweb.properties	2	su35e16/su35e17	4/1/2003	
29	mqipool.properties	12	su35e16/su35e17	4/1/2003	
30	mqstartstop.properties	2	su35e16/su35e17	4/1/2003	
31	saig	2	su35e16/su35e17	4/1/2003	
32	COD	1	su35e16/su35e17	4/1/2003	
33	prod	1	su35e16/su35e17	4/1/2003	
34	SAIGadapters.properties	1	su35e16/su35e17	4/1/2003	
35	SAIGcompressSecFile.properties	1	su35e16/su35e17	4/1/2003	
36	SAIGgetMessages.ini	1	su35e16/su35e17	4/1/2003	
37	SAIGmessageClass.properties	1	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
38	SAIGputMessages.ini	1	su35e16/su35e17	4/1/2003	
39	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
40	secfile.ini	1	su35e16/su35e17	4/1/2003	
41	schooltesting	1	su35e16/su35e17	4/1/2003	
42	SAIGadapters.properties	1	su35e16/su35e17	4/1/2003	
43	SAIGcompressSecFile.properties	1	su35e16/su35e17	4/1/2003	
44	SAIGgetMessages.ini	1	su35e16/su35e17	4/1/2003	
45	SAIGmessageClass.properties	1	su35e16/su35e17	4/1/2003	
46	SAIGputMessages.ini	1	su35e16/su35e17	4/1/2003	
47	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
48	secfile.ini	1	su35e16/su35e17	4/1/2003	
49	test	1	su35e16/su35e17	4/1/2003	
50	SAIGadapters.properties	1	su35e16/su35e17	4/1/2003	
51	SAIGcompressSecFile.properties	1	su35e16/su35e17	4/1/2003	
52	SAIGgetMessages.ini	1	su35e16/su35e17	4/1/2003	
53	SAIGmessageClass.properties	1	su35e16/su35e17	4/1/2003	
54	SAIGputMessages.ini	1	su35e16/su35e17	4/1/2003	
55	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
56	secfile.ini	1	su35e16/su35e17	4/1/2003	
57	LARS	2	su35e16/su35e17	4/1/2003	
58	prod	1	su35e16/su35e17	4/1/2003	
59	SAIGadapters.properties	1	su35e16/su35e17	4/1/2003	
60	SAIGcompressSecFile.properties	1	su35e16/su35e17	4/1/2003	
61	SAIGgetMessages.ini	1	su35e16/su35e17	4/1/2003	
62	SAIGmessageClass.properties	1	su35e16/su35e17	4/1/2003	
63	SAIGputMessages.ini	1	su35e16/su35e17	4/1/2003	
64	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
65	secfile.ini	1	su35e16/su35e17	4/1/2003	
66	test	1	su35e16/su35e17	4/1/2003	
67	SAIGadapter.properties	1	su35e16/su35e17	4/1/2003	
68	SAIGcompressSecFile.properties	1	su35e16/su35e17	4/1/2003	
69	SAIGgetMessages.ini	1	su35e16/su35e17	4/1/2003	
70	SAIGmessageClass.properties	1	su35e16/su35e17	4/1/2003	
71	SAIGputMessages.ini	1	su35e16/su35e17	4/1/2003	
72	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
73	secfile.ini	1	su35e16/su35e17	4/1/2003	
74	uat	2	su35e16/su35e17	4/1/2003	
75	SAIGcompressSecFile.properties	1	su35e16/su35e17	4/1/2003	
76	SAIGgetMessages.ini	2	su35e16/su35e17	4/1/2003	
77	SAIGmessageClass.properties	2	su35e16/su35e17	4/1/2003	
78	SAIGputMessages.ini	2	su35e16/su35e17	4/1/2003	
79	btradeagent.properties	2	su35e16/su35e17	4/1/2003	
80	secfile.ini	1	su35e16/su35e17	4/1/2003	
81	SAIGadapters.properties	1	su35e16/su35e17	4/1/2003	
82	SAIGcompressSecFile.properties	1	su35e16/su35e17	4/1/2003	
83	SAIGgetMessages.ini	2	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
84	SAIGmessageClass.properties	2	su35e16/su35e17	4/1/2003	
85	SAIGputMessages.ini	2	su35e16/su35e17	4/1/2003	
86	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
87	secfile.ini	1	su35e16/su35e17	4/1/2003	
88	secexit.ini	2	su35e16/su35e17	4/1/2003	
89	secexit.mqs	1	su35e16/su35e17	4/1/2003	
90	transform	5	su35e16/su35e17	4/1/2003	
91	x_LoggingConfig.xml	1	su35e16/su35e17	4/1/2003	
92	x_ResponseTemplates.xml	1	su35e16/su35e17	4/1/2003	
93	x_cr_response.xml	1	su35e16/su35e17	4/1/2003	
94	x_dl_batchEditMappings.xml	1	su35e16/su35e17	4/1/2003	
95	x_dl_booking_notif.xml	1	su35e16/su35e17	4/1/2003	
96	x_dl_changeCodeMappings.xml	1	su35e16/su35e17	4/1/2003	
97	x_dl_changeEditMappings.xml	1	su35e16/su35e17	4/1/2003	
98	x_dl_change_ack.xml	1	su35e16/su35e17	4/1/2003	
99	x_dl_creditdec_ack.xml	2	su35e16/su35e17	4/1/2003	
100	x_dl_plusOrigEditMappings.xml	1	su35e16/su35e17	4/1/2003	
101	x_dl_plusOrig_ack.xml	2	su35e16/su35e17	4/1/2003	
102	x_dl_pmttoserv_notif.xml	2	su35e16/su35e17	4/1/2003	
103	x_dl_pnote_ack.xml	1	su35e16/su35e17	4/1/2003	
104	x_dl_subunOrigEditMappings.xml	1	su35e16/su35e17	4/1/2003	
105	x_dl_subunOrig_ack.xml	2	su35e16/su35e17	4/1/2003	
106	x_dl_supDisbEditMappings.xml	1	su35e16/su35e17	4/1/2003	
107	x_dl_supDisb_ack.xml	2	su35e16/su35e17	4/1/2003	
108	x_pell_batchEditMappings.xml	1	su35e16/su35e17	4/1/2003	
109	x_pell_disbEditMappings.xml	3	su35e16/su35e17	4/1/2003	
110	x_pell_disb_ack.xml	3	su35e16/su35e17	4/1/2003	
111	x_pell_origEditMappings.xml	4	su35e16/su35e17	4/1/2003	
112	x_pell_orig_ack.xml	3	su35e16/su35e17	4/1/2003	
113	custom	1	su35e16/su35e17	4/1/2003	
114	C	2	su35e16/su35e17	4/1/2003	
115	inc	1	su35e16/su35e17	4/1/2003	
116	eaiLog.h	1	su35e16/su35e17	4/1/2003	
117	eaiMsgLog.h	1	su35e16/su35e17	4/1/2003	
118	samples	2	su35e16/su35e17	4/1/2003	
119	logamitest.cpp	1	su35e16/su35e17	4/1/2003	
120	logtest.cpp	1	su35e16/su35e17	4/1/2003	
121	logtest.o	1	su35e16/su35e17	4/1/2003	
122	src	2	su35e16/su35e17	4/1/2003	
123	ftf	1	su35e16/su35e17	4/1/2003	
124	EAIExits.c	2	su35e16/su35e17	4/1/2003	
125	EAIExitsNoLog.c	1	su35e16/su35e17	4/1/2003	
126	Makefile	1	su35e16/su35e17	4/1/2003	
127	ftfstatus	1	su35e16/su35e17	4/1/2003	
128	EAIStatus.c	2	su35e16/su35e17	4/1/2003	
129	Makefile	1	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
130	log	1	su35e16/su35e17	4/1/2003	
131	EAIMSGLOG.java	1	su35e16/su35e17	4/1/2003	
132	EAITESTLOG.java	1	su35e16/su35e17	4/1/2003	
133	MSG.java	1	su35e16/su35e17	4/1/2003	
134	Makefile	1	su35e16/su35e17	4/1/2003	
135	eaiLog.cpp	8	su35e16/su35e17	4/1/2003	
136	eaiLog.h	3	su35e16/su35e17	4/1/2003	
137	eaiMsgLog.h	1	su35e16/su35e17	4/1/2003	
138	eaiami.cpp	1	su35e16/su35e17	4/1/2003	
139	mqgetx	1	su35e16/su35e17	4/1/2003	
140	Makefile	1	su35e16/su35e17	4/1/2003	
141	amqsget.c	1	su35e16/su35e17	4/1/2003	
142	mqsave	1	su35e16/su35e17	4/1/2003	
143	Makefile	1	su35e16/su35e17	4/1/2003	
144	channel.c	1	su35e16/su35e17	4/1/2003	
145	mquutils.c	1	su35e16/su35e17	4/1/2003	
146	namelist.c	1	su35e16/su35e17	4/1/2003	
147	process.c	1	su35e16/su35e17	4/1/2003	
148	qmgr.c	1	su35e16/su35e17	4/1/2003	
149	queue.c	1	su35e16/su35e17	4/1/2003	
150	saveqmgr.c	1	su35e16/su35e17	4/1/2003	
151	saveqmgr.h	1	su35e16/su35e17	4/1/2003	
152	secexit	3	su35e16/su35e17	4/1/2003	
153	Makefile	3	su35e16/su35e17	4/1/2003	
154	secexit.c	3	su35e16/su35e17	4/1/2003	
155	secexit.h	1	su35e16/su35e17	4/1/2003	
156	timeutil	1	su35e16/su35e17	4/1/2003	
157	Makefile	1	su35e16/su35e17	4/1/2003	
158	getTimeStamp.c	2	su35e16/su35e17	4/1/2003	
159	java	1	su35e16/su35e17	4/1/2003	
160	samples	3	su35e16/su35e17	4/1/2003	
161	BorrowerValSendReceive.java	2	su35e16/su35e17	4/1/2003	
162	CreditCheckSendReceive.java	2	su35e16/su35e17	4/1/2003	
163	EAIMSGLOG.java	1	su35e16/su35e17	4/1/2003	
164	EMPNTESTSTUB.java	1	su35e16/su35e17	4/1/2003	
165	EMPNX.java	1	su35e16/su35e17	4/1/2003	
166	EMPNXMAIN.java	1	su35e16/su35e17	4/1/2003	
167	Endorsersend.java	2	su35e16/su35e17	4/1/2003	
168	MSG.java	1	su35e16/su35e17	4/1/2003	
169	Makefile	1	su35e16/su35e17	4/1/2003	
170	Pnotesend.java	1	su35e16/su35e17	4/1/2003	
171	bv.txt	1	su35e16/su35e17	4/1/2003	
172	cred.txt	1	su35e16/su35e17	4/1/2003	
173	endorser.txt	1	su35e16/su35e17	4/1/2003	
174	lowebR2	1	su35e16/su35e17	4/1/2003	
175	BorrowerValSendReceive.java	2	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
176	CreditCheckSendReceive.java	2	su35e16/su35e17	4/1/2003	
177	Endorsersend.java	2	su35e16/su35e17	4/1/2003	
178	Pnotesend.java	2	su35e16/su35e17	4/1/2003	
179	bv.sh	2	su35e16/su35e17	4/1/2003	
180	bv.txt	1	su35e16/su35e17	4/1/2003	
181	cred.sh	2	su35e16/su35e17	4/1/2003	
182	cred.txt	2	su35e16/su35e17	4/1/2003	
183	endorser.sh	2	su35e16/su35e17	4/1/2003	
184	endorser.txt	1	su35e16/su35e17	4/1/2003	
185	pnote.sh	2	su35e16/su35e17	4/1/2003	
186	pnote.txt	1	su35e16/su35e17	4/1/2003	
187	lowebR2Perf	1	su35e16/su35e17	4/1/2003	
188	BVthread.java	2	su35e16/su35e17	4/1/2003	
189	CREDthread.java	2	su35e16/su35e17	4/1/2003	
190	ENDthread.java	2	su35e16/su35e17	4/1/2003	
191	LOWebPerf.java	2	su35e16/su35e17	4/1/2003	
192	LOWebPerf.sh	2	su35e16/su35e17	4/1/2003	
193	PNthread.java	2	su35e16/su35e17	4/1/2003	
194	bv.txt	1	su35e16/su35e17	4/1/2003	
195	cred.txt	2	su35e16/su35e17	4/1/2003	
196	endorser.txt	1	su35e16/su35e17	4/1/2003	
197	pnote.txt	1	su35e16/su35e17	4/1/2003	
198	pnote.txt	1	su35e16/su35e17	4/1/2003	
199	src	7	su35e16/su35e17	4/1/2003	
200	archivefiles	5	su35e16/su35e17	4/1/2003	
201	ArchiveFiles.java	3	su35e16/su35e17	4/1/2003	
202	EAIFileHandle.java	1	su35e16/su35e17	4/1/2003	
203	EAIFileHandleList.java	1	su35e16/su35e17	4/1/2003	
204	EAIFileHandleListException.java	1	su35e16/su35e17	4/1/2003	
205	GZIPTimestampInputStream.java	1	su35e16/su35e17	4/1/2003	
206	GZIPTimestampOutputStream.java	1	su35e16/su35e17	4/1/2003	
207	GetOpt.java	1	su35e16/su35e17	4/1/2003	
208	GrepDirUsingFile.java	1	su35e16/su35e17	4/1/2003	
209	Makefile	6	su35e16/su35e17	4/1/2003	
210	PurgeFiles.java	4	su35e16/su35e17	4/1/2003	
211	RestoreFiles.java	1	su35e16/su35e17	4/1/2003	
212	WildcardFilter.java	1	su35e16/su35e17	4/1/2003	
213	cleanipcs	1	su35e16/su35e17	4/1/2003	
214	Cleanipcs.java	1	su35e16/su35e17	4/1/2003	
215	GetOpt.java	1	su35e16/su35e17	4/1/2003	
216	Makefile	1	su35e16/su35e17	4/1/2003	
217	dilog	1	su35e16/su35e17	4/1/2003	
218	DIControl.java	1	su35e16/su35e17	4/1/2003	
219	DIDetail.java	1	su35e16/su35e17	4/1/2003	
220	DILogInsert.java	1	su35e16/su35e17	4/1/2003	
221	DILogSelect.java	1	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
222	DILogSelectEngine.java	1	su35e16/su35e17	4/1/2003	
223	dlos	5	su35e16/su35e17	4/1/2003	
224	BVReply.java	2	su35e16/su35e17	4/1/2003	
225	BVRequest.java	2	su35e16/su35e17	4/1/2003	
226	EndorserTransform.java	2	su35e16/su35e17	4/1/2003	
227	Makefile	7	su35e16/su35e17	4/1/2003	
228	NameValueTransform.java	6	su35e16/su35e17	4/1/2003	
229	PnoteTransform.java	6	su35e16/su35e17	4/1/2003	
230	eaiengine	4	su35e16/su35e17	4/1/2003	
231	DIControl.java	1	su35e16/su35e17	4/1/2003	
232	DIDetail.java	1	su35e16/su35e17	4/1/2003	
233	DILogInsert.java	7	su35e16/su35e17	4/1/2003	
234	DILogSelect.java	1	su35e16/su35e17	4/1/2003	
235	DILogSelectEngine.java	1	su35e16/su35e17	4/1/2003	
236	DIPagerEngine.java	6	su35e16/su35e17	4/1/2003	
237	EAIProcessInitQ.java	11	su35e16/su35e17	4/1/2003	
238	EAIProcessQ.java	4	su35e16/su35e17	4/1/2003	
239	EAIProcessQFactory.java	9	su35e16/su35e17	4/1/2003	
240	EAIProcessQStandard.java	15	su35e16/su35e17	4/1/2003	
241	EAIRollbackException.java	4	su35e16/su35e17	4/1/2003	
242	EAITransformEngine.java	4	su35e16/su35e17	4/1/2003	
243	EAITransformEngineTest.java	5	su35e16/su35e17	4/1/2003	
244	GetOpt.java	2	su35e16/su35e17	4/1/2003	
245	Makefile	7	su35e16/su35e17	4/1/2003	
246	PageDispatch.java	3	su35e16/su35e17	4/1/2003	
247	backend	2	su35e16/su35e17	4/1/2003	
248	Makefile	1	su35e16/su35e17	4/1/2003	
249	RealTimeCRStub.java	2	su35e16/su35e17	4/1/2003	
250	stubtemplate	1	su35e16/su35e17	4/1/2003	
251	StubTemplate.java	1	su35e16/su35e17	4/1/2003	
252	test.properties	6	su35e16/su35e17	4/1/2003	
253	fms	2	su35e16/su35e17	4/1/2003	
254	CODXFinancial_v401	1	su35e16/su35e17	4/1/2003	
255	CODXResponse_v401	1	su35e16/su35e17	4/1/2003	
256	MQFMS.java	16	su35e16/su35e17	4/1/2003	
257	MQFMS.java.NoGaps.20030407	4	su35e16/su35e17	4/1/2003	
258	Makefile	2	su35e16/su35e17	4/1/2003	
259	frontend	5	su35e16/su35e17	4/1/2003	
260	EAI.java	7	su35e16/su35e17	4/1/2003	
261	EAIException.java	5	su35e16/su35e17	4/1/2003	
262	EAIFactory.java	8	su35e16/su35e17	4/1/2003	
263	EAIMQQueueManager.java	6	su35e16/su35e17	4/1/2003	
264	EAIMQQueueManagerConnections.java	25	su35e16/su35e17	4/1/2003	
265	EAIMQQueueManagerFactory.java	9	su35e16/su35e17	4/1/2003	
266	EAIMQQueueManagerHolder.java	9	su35e16/su35e17	4/1/2003	
267	EAImqi.java	12	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
268	EAlpool.java	18	su35e16/su35e17	4/1/2003	
269	IEAI.java	6	su35e16/su35e17	4/1/2003	
270	Makefile	7	su35e16/su35e17	4/1/2003	
271	javalog	3	su35e16/su35e17	4/1/2003	
272	EAILog.java	4	su35e16/su35e17	4/1/2003	
273	Makefile	4	su35e16/su35e17	4/1/2003	
274	peps	2	su35e16/su35e17	4/1/2003	
275	GetOpt.java	1	su35e16/su35e17	4/1/2003	
276	Makefile	3	su35e16/su35e17	4/1/2003	
277	PEPSDeltaBuild.java	2	su35e16/su35e17	4/1/2003	
278	PEPSShieldTSYS.java	1	su35e16/su35e17	4/1/2003	
279	SFAException.java	1	su35e16/su35e17	4/1/2003	
280	saig	2	su35e16/su35e17	4/1/2003	
281	FileModifiedSince.java	1	su35e16/su35e17	4/1/2003	
282	Makefile	2	su35e16/su35e17	4/1/2003	
283	SAIGgetMessages.java	1	su35e16/su35e17	4/1/2003	
284	SAIGhelpers.java	1	su35e16/su35e17	4/1/2003	
285	SAIGputMessages.java	2	su35e16/su35e17	4/1/2003	
286	transform	1	su35e16/su35e17	4/1/2003	
287	ConfigLoader.java	1	su35e16/su35e17	4/1/2003	
288	EmptyHandler.java	1	su35e16/su35e17	4/1/2003	
289	FFRowField.java	1	su35e16/su35e17	4/1/2003	
290	FFRowTemplate.java	1	su35e16/su35e17	4/1/2003	
291	FFTTemplateMgr.java	1	su35e16/su35e17	4/1/2003	
292	FormatUtils.java	1	su35e16/su35e17	4/1/2003	
293	Makefile	1	su35e16/su35e17	4/1/2003	
294	RespPreScanner.java	1	su35e16/su35e17	4/1/2003	
295	RespSAXHandler.java	1	su35e16/su35e17	4/1/2003	
296	RespScanner.java	1	su35e16/su35e17	4/1/2003	
297	RespTransformer.java	3	su35e16/su35e17	4/1/2003	
298	RespTransformerCR.java	1	su35e16/su35e17	4/1/2003	
299	RespTransformerDLC.java	2	su35e16/su35e17	4/1/2003	
300	ResponseProcessor.java	1	su35e16/su35e17	4/1/2003	
301	Transformer.java	1	su35e16/su35e17	4/1/2003	
302	util	4	su35e16/su35e17	4/1/2003	
303	AmqsBcgRestore.java	1	su35e16/su35e17	4/1/2003	
304	GetOpt.java	1	su35e16/su35e17	4/1/2003	
305	SendMail.java	1	su35e16/su35e17	4/1/2003	
306	middleware	1	su35e16/su35e17	4/1/2003	
307	DI	2	su35e16/su35e17	4/1/2003	
308	MQSI	3	su35e16/su35e17	4/1/2003	
309	CODFMSV1.mrp	1	su35e16/su35e17	4/1/2003	
310	DLOS_FLOWS_MAY21_2002.zip	1	su35e16/su35e17	4/1/2003	
311	DLOS_FLOWS_PROD_JUNE3.zip	1	su35e16/su35e17	4/1/2003	
312	DLOS_FLOW_MAY20_2002.zip	1	su35e16/su35e17	4/1/2003	
313	DLOS_MRM.OUT.mrp	1	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
314	DLOS_MRM_OUT.mrp	1	su35e16/su35e17	4/1/2003	
315	DLOS_MessageSet.xml	1	su35e16/su35e17	4/1/2003	
316	FMSMessageFlows.zip	1	su35e16/su35e17	4/1/2003	
317	FMS_DLOS_MessageFlows.xml.Z	1	su35e16/su35e17	4/1/2003	
318	FMS_MessageSet.xml	1	su35e16/su35e17	4/1/2003	
319	MQSI_Prod_Flows_Mar17_2003.gz	1	su35e16/su35e17	4/1/2003	
320	cod_fms_v1.1_mrm_bkp_mar17.mrp.gz	1	su35e16/su35e17	4/1/2003	
321	dlos_message_flows.zip	1	su35e16/su35e17	4/1/2003	
322	dlos_message_flows_01_22_2002.ZIP	1	su35e16/su35e17	4/1/2003	
323	dlos_message_flows_dec20.zip	1	su35e16/su35e17	4/1/2003	
324	dlos_mrm_bkp_mar17.mrp.gz	1	su35e16/su35e17	4/1/2003	
325	dlos_mrm_feb12_2002.mrp	1	su35e16/su35e17	4/1/2003	
326	dlos_mrm_may20_2002.mrp	1	su35e16/su35e17	4/1/2003	
327	dlos_mrm_prod_fix.mrp	1	su35e16/su35e17	4/1/2003	
328	dlos_mrm_prod_jun3.mrp	1	su35e16/su35e17	4/1/2003	
329	dlos_r1.1_flows.zip	1	su35e16/su35e17	4/1/2003	
330	fms_unpaid_mrm_bkp_mar17.mrp.gz	1	su35e16/su35e17	4/1/2003	
331	MQSeries	2	su35e16/su35e17	4/1/2003	
332	eCBtoFMS	1	su35e16/su35e17	4/1/2003	
333	EAIP1.sh	1	su35e16/su35e17	4/1/2003	
334	EAIP2.sh	1	su35e16/su35e17	4/1/2003	
335	ECBSP1.sh	1	su35e16/su35e17	4/1/2003	
336	FMSP1.sh	1	su35e16/su35e17	4/1/2003	
337	eCBFMSInterfaceHand-over.doc	1	su35e16/su35e17	4/1/2003	
338	scripts	8	su35e16/su35e17	4/1/2003	
339	2301.sh	1	su35e16/su35e17	4/1/2003	
340	2302.sh	1	su35e16/su35e17	4/1/2003	
341	2401.sh	1	su35e16/su35e17	4/1/2003	
342	2402.sh	1	su35e16/su35e17	4/1/2003	
343	EAIDLOSEngine.sh	2	su35e16/su35e17	4/1/2003	
344	EAIEngine.sh	1	su35e16/su35e17	4/1/2003	
345	FMSInsert	3	su35e16/su35e17	4/1/2003	
346	FMSSelect.sh	1	su35e16/su35e17	4/1/2003	
347	FMSfilearchive.sh	1	su35e16/su35e17	4/1/2003	
348	PEPSCOMBINED.sh	1	su35e16/su35e17	4/1/2003	
349	PEPSDELTA.sh	1	su35e16/su35e17	4/1/2003	
350	PEPSFULLREFRESH.sh	1	su35e16/su35e17	4/1/2003	
351	PEPSSHIELD.sh	1	su35e16/su35e17	4/1/2003	
352	PEPSSHIELDFULLREFRESH.sh	1	su35e16/su35e17	4/1/2003	
353	amienvlist.sh	1	su35e16/su35e17	4/1/2003	
354	codXform.sh	1	su35e16/su35e17	4/1/2003	
355	diclientstartstop.sh	3	su35e16/su35e17	4/1/2003	
356	dirmonstart.sh	1	su35e16/su35e17	4/1/2003	
357	distartstop.sh	5	su35e16/su35e17	4/1/2003	
358	eaiclean.sh	1	su35e16/su35e17	4/1/2003	
359	eaidata_setup.sh	1	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
360	fmsenvlist.sh	2	su35e16/su35e17	4/1/2003	
361	ftf	4	su35e16/su35e17	4/1/2003	
362	CODsendschooldestfile	1	su35e16/su35e17	4/1/2003	
363	CPSaar	1	su35e16/su35e17	4/1/2003	
364	DATADIR.sh	1	su35e16/su35e17	4/1/2003	
365	DLSSdaily	1	su35e16/su35e17	4/1/2003	
366	DLSSweekly	1	su35e16/su35e17	4/1/2003	
367	FARSScript	1	su35e16/su35e17	4/1/2003	
368	FARSSendfile	3	su35e16/su35e17	4/1/2003	
369	Fpjobtrigger.sh	1	su35e16/su35e17	4/1/2003	
370	Fpsendfile.sh	1	su35e16/su35e17	4/1/2003	
371	NSLDSsendIB.sh	1	su35e16/su35e17	4/1/2003	
372	NSLDSsendLD.sh	1	su35e16/su35e17	4/1/2003	
373	NSLDSsendSA.sh	1	su35e16/su35e17	4/1/2003	
374	PEPSfull	1	su35e16/su35e17	4/1/2003	
375	PEPSmaintenance	1	su35e16/su35e17	4/1/2003	
376	Pepsdeltasend.sh	1	su35e16/su35e17	4/1/2003	
377	Pepsfullfile.sh	6	su35e16/su35e17	4/1/2003	
378	SAIGjob.sh	3	su35e16/su35e17	4/1/2003	
379	eCBFMSobexpTransfer.sh	2	su35e16/su35e17	4/1/2003	
380	eCBFMSpbsTransfer.sh	2	su35e16/su35e17	4/1/2003	
381	eCBFMSutclTransfer.sh	2	su35e16/su35e17	4/1/2003	
382	eCBsendfile.sh	1	su35e16/su35e17	4/1/2003	
383	eZAuditSendFile.sh	3	su35e16/su35e17	4/1/2003	
384	ftfstartprocess	1	su35e16/su35e17	4/1/2003	
385	sendCRfile	1	su35e16/su35e17	4/1/2003	
386	sendLRfileDLC	1	su35e16/su35e17	4/1/2003	
387	sendLRfileDLD	1	su35e16/su35e17	4/1/2003	
388	sendLRfileDLO	1	su35e16/su35e17	4/1/2003	
389	sendLRfilePLD	1	su35e16/su35e17	4/1/2003	
390	sendLRfilePLO	1	su35e16/su35e17	4/1/2003	
391	timestampjob.sh	1	su35e16/su35e17	4/1/2003	
392	ftfenvlist.sh	6	su35e16/su35e17	4/1/2003	
393	lowebenvlist.sh	1	su35e16/su35e17	4/1/2003	
394	makeOS-sparc.inc	1	su35e16/su35e17	4/1/2003	
395	makeOS.inc	1	su35e16/su35e17	4/1/2003	
396	makeglobal.inc	13	su35e16/su35e17	4/1/2003	
397	mqipcrm.ksh	1	su35e16/su35e17	4/1/2003	
398	mqsistartstop.sh	6	su35e16/su35e17	4/1/2003	
399	mqstartstop.sh	7	su35e16/su35e17	4/1/2003	
400	mqtriggerstart.sh	2	su35e16/su35e17	4/1/2003	
401	ofile1	1	su35e16/su35e17	4/1/2003	
402	profile.txt	1	su35e16/su35e17	4/1/2003	
403	saig	4	su35e16/su35e17	4/1/2003	
404	.profile	1	su35e16/su35e17	4/1/2003	
405	COD	1	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
406	prod	1	su35e16/su35e17	4/1/2003	
407	CODESTIN.sh	1	su35e16/su35e17	4/1/2003	
408	COMRECIN.sh	1	su35e16/su35e17	4/1/2003	
409	CRTESTIN.sh	1	su35e16/su35e17	4/1/2003	
410	DEPF03IN.sh	1	su35e16/su35e17	4/1/2003	
411	DESC03IN.sh	1	su35e16/su35e17	4/1/2003	
412	DESD03IN.sh	1	su35e16/su35e17	4/1/2003	
413	DESF03IN.sh	1	su35e16/su35e17	4/1/2003	
414	DTPF03IN.sh	1	su35e16/su35e17	4/1/2003	
415	DTSC03IN.sh	1	su35e16/su35e17	4/1/2003	
416	DTSD03IN.sh	1	su35e16/su35e17	4/1/2003	
417	DTSF03IN.sh	1	su35e16/su35e17	4/1/2003	
418	PGDR03IN.sh	1	su35e16/su35e17	4/1/2003	
419	PGOR03IN.sh	1	su35e16/su35e17	4/1/2003	
420	PTDR03IN.sh	1	su35e16/su35e17	4/1/2003	
421	PTOR03IN.sh	1	su35e16/su35e17	4/1/2003	
422	UNKNOWN.sh	1	su35e16/su35e17	4/1/2003	
423	addMessage.ksh	1	su35e16/su35e17	4/1/2003	
424	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
425	getMessages.sh	1	su35e16/su35e17	4/1/2003	
426	putMessage.ksh	1	su35e16/su35e17	4/1/2003	
427	putWrapper.ksh	1	su35e16/su35e17	4/1/2003	
428	schoolstesting	1	su35e16/su35e17	4/1/2003	
429	CODESTIN.sh	1	su35e16/su35e17	4/1/2003	
430	COMRECIN.sh	1	su35e16/su35e17	4/1/2003	
431	CRTESTIN.sh	1	su35e16/su35e17	4/1/2003	
432	DEPF03IN.sh	1	su35e16/su35e17	4/1/2003	
433	DESC03IN.sh	1	su35e16/su35e17	4/1/2003	
434	DESD03IN.sh	1	su35e16/su35e17	4/1/2003	
435	DESF03IN.sh	1	su35e16/su35e17	4/1/2003	
436	DTPF03IN.sh	1	su35e16/su35e17	4/1/2003	
437	DTSC03IN.sh	1	su35e16/su35e17	4/1/2003	
438	DTSD03IN.sh	1	su35e16/su35e17	4/1/2003	
439	DTSF03IN.sh	1	su35e16/su35e17	4/1/2003	
440	PGDR03IN.sh	1	su35e16/su35e17	4/1/2003	
441	PGOR03IN.sh	1	su35e16/su35e17	4/1/2003	
442	PTDR03IN.sh	1	su35e16/su35e17	4/1/2003	
443	PTOR03IN.sh	1	su35e16/su35e17	4/1/2003	
444	RPTESTIN.sh	1	su35e16/su35e17	4/1/2003	
445	UNKNOWN.sh	1	su35e16/su35e17	4/1/2003	
446	addMessage.ksh	1	su35e16/su35e17	4/1/2003	
447	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
448	getMessages.sh	1	su35e16/su35e17	4/1/2003	
449	putMessage.ksh	1	su35e16/su35e17	4/1/2003	
450	putWrapper.ksh	1	su35e16/su35e17	4/1/2003	
451	test	2	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
452	CODESTIN.sh	1	su35e16/su35e17	4/1/2003	
453	COMRECIN.sh	1	su35e16/su35e17	4/1/2003	
454	CRTESTIN.sh	1	su35e16/su35e17	4/1/2003	
455	DEPF03IN.sh	1	su35e16/su35e17	4/1/2003	
456	DESC03IN.sh	1	su35e16/su35e17	4/1/2003	
457	DESD03IN.sh	1	su35e16/su35e17	4/1/2003	
458	DESF03IN.sh	1	su35e16/su35e17	4/1/2003	
459	DTPF03IN.sh	1	su35e16/su35e17	4/1/2003	
460	DTSC03IN.sh	1	su35e16/su35e17	4/1/2003	
461	DTSD03IN.sh	1	su35e16/su35e17	4/1/2003	
462	DTSF03IN.sh	1	su35e16/su35e17	4/1/2003	
463	PGDR03IN.sh	1	su35e16/su35e17	4/1/2003	
464	PGOR03IN.sh	1	su35e16/su35e17	4/1/2003	
465	PTDR03IN.sh	1	su35e16/su35e17	4/1/2003	
466	PTOR03IN.sh	1	su35e16/su35e17	4/1/2003	
467	PTRQ03IN.sh	1	su35e16/su35e17	4/1/2003	
468	UNKNOWN.sh	1	su35e16/su35e17	4/1/2003	
469	addMessage.ksh	1	su35e16/su35e17	4/1/2003	
470	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
471	getMessages.sh	1	su35e16/su35e17	4/1/2003	
472	putMessage.ksh	1	su35e16/su35e17	4/1/2003	
473	putMessage.sh	1	su35e16/su35e17	4/1/2003	
474	putWrapper.ksh	1	su35e16/su35e17	4/1/2003	
475	CODESTIN.sh	2	su35e16/su35e17	4/1/2003	
476	COMRECIN.sh	2	su35e16/su35e17	4/1/2003	
477	COMRECOP.sh	1	su35e16/su35e17	4/1/2003	
478	CRTESTIN.sh	1	su35e16/su35e17	4/1/2003	
479	DEPF03IN.sh	1	su35e16/su35e17	4/1/2003	
480	DESC03IN.sh	1	su35e16/su35e17	4/1/2003	
481	DESD03IN.sh	1	su35e16/su35e17	4/1/2003	
482	DESF03IN.sh	1	su35e16/su35e17	4/1/2003	
483	DTPF03IN.sh	1	su35e16/su35e17	4/1/2003	
484	DTSC03IN.sh	1	su35e16/su35e17	4/1/2003	
485	DTSD03IN.sh	1	su35e16/su35e17	4/1/2003	
486	DTSF03IN.sh	1	su35e16/su35e17	4/1/2003	
487	LARS	1	su35e16/su35e17	4/1/2003	
488	prod	2	su35e16/su35e17	4/1/2003	
489	FMSMLBIN.sh	2	su35e16/su35e17	4/1/2003	
490	LARSIVIN.sh	2	su35e16/su35e17	4/1/2003	
491	UNKNOWN.sh	2	su35e16/su35e17	4/1/2003	
492	addMessage.ksh	1	su35e16/su35e17	4/1/2003	
493	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
494	getMessages.sh	1	su35e16/su35e17	4/1/2003	
495	putMessage.ksh	1	su35e16/su35e17	4/1/2003	
496	putWrapper.ksh	1	su35e16/su35e17	4/1/2003	
497	uat	2	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
498	TFMSLBIN.sh	1	su35e16/su35e17	4/1/2003	
499	TLARSIIN.sh	2	su35e16/su35e17	4/1/2003	
500	UNKNOWN.sh	1	su35e16/su35e17	4/1/2003	
501	addMessage.ksh	1	su35e16/su35e17	4/1/2003	
502	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
503	getMessages.sh	1	su35e16/su35e17	4/1/2003	
504	putMessage.ksh	1	su35e16/su35e17	4/1/2003	
505	putWrapper.ksh	1	su35e16/su35e17	4/1/2003	
506	MELLON	1	su35e16/su35e17	4/1/2003	
507	prod	1	su35e16/su35e17	4/1/2003	
508	uat	1	su35e16/su35e17	4/1/2003	
509	TFMSLBIN.sh	1	su35e16/su35e17	4/1/2003	
510	TLARSIIN.sh	1	su35e16/su35e17	4/1/2003	
511	UNKNOWN.sh	1	su35e16/su35e17	4/1/2003	
512	addMessage.ksh	1	su35e16/su35e17	4/1/2003	
513	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
514	getMessages.sh	1	su35e16/su35e17	4/1/2003	
515	putMessage.ksh	1	su35e16/su35e17	4/1/2003	
516	putWrapper.ksh	1	su35e16/su35e17	4/1/2003	
517	PGDR03IN.sh	1	su35e16/su35e17	4/1/2003	
518	PGOR03IN.sh	1	su35e16/su35e17	4/1/2003	
519	PTDR03IN.sh	1	su35e16/su35e17	4/1/2003	
520	PTOR03IN.sh	1	su35e16/su35e17	4/1/2003	
521	UNKNOWN.sh	1	su35e16/su35e17	4/1/2003	
522	addMessage.ksh	1	su35e16/su35e17	4/1/2003	
523	btradeagent.properties	1	su35e16/su35e17	4/1/2003	
524	getMessages.sh	2	su35e16/su35e17	4/1/2003	
525	putMessage.ksh	2	su35e16/su35e17	4/1/2003	
526	putMessage.sh	1	su35e16/su35e17	4/1/2003	
527	putWrapper.ksh	1	su35e16/su35e17	4/1/2003	
528	sendErrorFileToCOD	2	su35e16/su35e17	4/1/2003	
529	test.xml	3	su35e16/su35e17	4/1/2003	
530	util	2	su35e16/su35e17	4/1/2003	
531	connectorapi	1	su35e16/su35e17	4/1/2003	
532	ConnectorAPI.jar	1	su35e16/su35e17	4/1/2003	
533	javamail1_3	1	su35e16/su35e17	4/1/2003	
534	activation.jar	2	su35e16/su35e17	4/1/2003	
535	mail.jar	2	su35e16/su35e17	4/1/2003	
536	mailapi.jar	2	su35e16/su35e17	4/1/2003	
537	smtp.jar	1	su35e16/su35e17	4/1/2003	
538	jdom	1	su35e16/su35e17	4/1/2003	
539	jdom-b7.tar.gz	1	su35e16/su35e17	4/1/2003	
540	jdom-jdk11.jar	1	su35e16/su35e17	4/1/2003	
541	jdom.jar	1	su35e16/su35e17	4/1/2003	
542	ncsjni	1	su35e16/su35e17	4/1/2003	
543	ncsjni110901.tar	1	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
544	oracle	1	su35e16/su35e17	4/1/2003	
545	classes12.zip	1	su35e16/su35e17	4/1/2003	
546	protomatter	1	su35e16/su35e17	4/1/2003	
547	protomatter-1.1.5.jar	1	su35e16/su35e17	4/1/2003	
548	protomatter-javadoc-1.1.5.jar	1	su35e16/su35e17	4/1/2003	
549	xml	2	su35e16/su35e17	4/1/2003	
550	Xerces-J-bin.1.4.4.tar.gz	1	su35e16/su35e17	4/1/2003	
551	xerces.jar	1	su35e16/su35e17	4/1/2003	
552	xercesSamples.jar	1	su35e16/su35e17	4/1/2003	
553	xml4j.jar	1	su35e16/su35e17	4/1/2003	
554	test	2	su35e16/su35e17	4/1/2003	
555	COD	1	su35e16/su35e17	4/1/2003	
556	scripts	1	su35e16/su35e17	4/1/2003	
557	changetext.ex.cmd	1	su35e16/su35e17	4/1/2003	
558	changetext.ksh	1	su35e16/su35e17	4/1/2003	
559	chk_diff.ksh	1	su35e16/su35e17	4/1/2003	
560	codXform.ksh	1	su35e16/su35e17	4/1/2003	
561	codXformB.ksh	1	su35e16/su35e17	4/1/2003	
562	renamefiles.ksh	1	su35e16/su35e17	4/1/2003	
563	xform_test_env_setup.ksh	1	su35e16/su35e17	4/1/2003	
564	COD-R2.0	2	su35e16/su35e17	4/1/2003	
565	config	1	su35e16/su35e17	4/1/2003	
566	transform	2	su35e16/su35e17	4/1/2003	
567	x_LoggingConfig.xml	1	su35e16/su35e17	4/1/2003	
568	x_ResponseTemplates.xml	1	su35e16/su35e17	4/1/2003	
569	x_cr_response.xml	2	su35e16/su35e17	4/1/2003	
570	x_dl_batchEditMappings_0203.xml	2	su35e16/su35e17	4/1/2003	
571	x_dl_batchEditMappings_0304.xml	3	su35e16/su35e17	4/1/2003	
572	x_dl_booking_notif_0203.xml	2	su35e16/su35e17	4/1/2003	
573	x_dl_booking_notif_0304.xml	2	su35e16/su35e17	4/1/2003	
574	x_dl_changeCodeMappings_0203.xml	2	su35e16/su35e17	4/1/2003	
575	x_dl_changeCodeMappings_0304.xml	2	su35e16/su35e17	4/1/2003	
576	x_dl_changeEditMappings_0203.xml	2	su35e16/su35e17	4/1/2003	
577	x_dl_changeEditMappings_0304.xml	2	su35e16/su35e17	4/1/2003	
578	x_dl_change_ack_0203.xml	2	su35e16/su35e17	4/1/2003	
579	x_dl_change_ack_0304.xml	2	su35e16/su35e17	4/1/2003	
580	x_dl_creditdec_ack_0203.xml	2	su35e16/su35e17	4/1/2003	
581	x_dl_creditdec_ack_0304.xml	2	su35e16/su35e17	4/1/2003	
582	x_dl_plusOrigEditMappings_0203.xml	2	su35e16/su35e17	4/1/2003	
583	x_dl_plusOrigEditMappings_0304.xml	2	su35e16/su35e17	4/1/2003	
584	x_dl_plusOrig_ack_0203.xml	2	su35e16/su35e17	4/1/2003	
585	x_dl_plusOrig_ack_0304.xml	2	su35e16/su35e17	4/1/2003	
586	x_dl_pmttoserv_notif_0203.xml	2	su35e16/su35e17	4/1/2003	
587	x_dl_pmttoserv_notif_0304.xml	2	su35e16/su35e17	4/1/2003	
588	x_dl_pnote_ack_0203.xml	2	su35e16/su35e17	4/1/2003	
589	x_dl_pnote_ack_0304.xml	2	su35e16/su35e17	4/1/2003	



Control #	Filename	Revision	Location	Date	Changes
590	x_dl_subunOrigEditMappings_0203.xml	2	su35e16/su35e17	4/1/2003	
591	x_dl_subunOrigEditMappings_0304.xml	2	su35e16/su35e17	4/1/2003	
592	x_dl_subunOrig_ack_0203.xml	2	su35e16/su35e17	4/1/2003	
593	x_dl_subunOrig_ack_0304.xml	2	su35e16/su35e17	4/1/2003	
594	x_dl_supDisbEditMappings_0203.xml	2	su35e16/su35e17	4/1/2003	
595	x_dl_supDisbEditMappings_0304.xml	2	su35e16/su35e17	4/1/2003	
596	x_dl_supDisb_ack_0203.xml	2	su35e16/su35e17	4/1/2003	
597	x_dl_supDisb_ack_0304.xml	2	su35e16/su35e17	4/1/2003	
598	x_pell_batchEditMappings_0203.xml	2	su35e16/su35e17	4/1/2003	
599	x_pell_batchEditMappings_0304.xml	2	su35e16/su35e17	4/1/2003	
600	x_pell_disbEditMappings_0203.xml	2	su35e16/su35e17	4/1/2003	
601	x_pell_disbEditMappings_0304.xml	1	su35e16/su35e17	4/1/2003	
602	x_pell_disb_ack_0203.xml	2	su35e16/su35e17	4/1/2003	
603	x_pell_disb_ack_0304.xml	2	su35e16/su35e17	4/1/2003	
604	x_pell_origEditMappings_0203.xml	2	su35e16/su35e17	4/1/2003	
605	x_pell_origEditMappings_0304.xml	2	su35e16/su35e17	4/1/2003	
606	x_pell_orig_ack_0203.xml	4	su35e16/su35e17	4/1/2003	
607	x_pell_orig_ack_0304.xml	4	su35e16/su35e17	4/1/2003	
608	scripts	1	su35e16/su35e17	4/1/2003	
609	codXform.sh	1	su35e16/su35e17	4/1/2003	
610	ftf	1	su35e16/su35e17	4/1/2003	
611	SAIGjob.sh	3	su35e16/su35e17	4/1/2003	
612	src	1	su35e16/su35e17	4/1/2003	
613	transform	1	su35e16/su35e17	4/1/2003	
614	ConfigLoader.java	1	su35e16/su35e17	4/1/2003	
615	EmptyHandler.java	1	su35e16/su35e17	4/1/2003	
616	FFRowField.java	1	su35e16/su35e17	4/1/2003	
617	FFRowTemplate.java	1	su35e16/su35e17	4/1/2003	
618	FFTTemplateMgr.java	1	su35e16/su35e17	4/1/2003	
619	FormatUtils.java	1	su35e16/su35e17	4/1/2003	
620	Makefile	1	su35e16/su35e17	4/1/2003	
621	RespPreScanner.java	1	su35e16/su35e17	4/1/2003	
622	RespSAXHandler.java	1	su35e16/su35e17	4/1/2003	
623	RespScanner.java	1	su35e16/su35e17	4/1/2003	
624	RespTransformer.java	1	su35e16/su35e17	4/1/2003	
625	RespTransformerCR.java	1	su35e16/su35e17	4/1/2003	
626	RespTransformerDLC.java	1	su35e16/su35e17	4/1/2003	
627	ResponseProcessor.java	1	su35e16/su35e17	4/1/2003	
628	Transformer.java	1	su35e16/su35e17	4/1/2003	
629	version.txt	1	su35e16/su35e17	4/1/2003	



APPENDIX B: ClearCase UNIX Command Line Interface Commands

The most commonly used ClearCase UNIX commands are listed below:

Execute ClearCase Profile

```
. /cc/vobstore/admin/scripts/ccenvlist.sh
```

Using Help

```
cleartool help ClearCase_command  
cleartool man -g
```

Create View

```
cleartool mkview -tag name_view /cc/viewstore/name.vws
```

Set View

```
cleartool setview name_view
```

Check Out Element

```
cleartool checkout -c "comment" element_name  
cleartool co -nc element_name
```

Check In Element

```
cleartool checkin -c "comment" element_name
```

Uncheck Out Element

```
cleartool uncheckout element_name  
cleartool unco element_name
```

Create Element

```
cleartool mkelem -c "comment" element_name
```

Make Directory

```
cleartool mkdir -c "comment" dir_name
```

List Elements Checked Out

```
cleartool lscheckout -all  
cleartool lsco -all
```

Branching

```
cleartool catcs – modify the config spec using the Rational ClearCase Administrator's Guide
```



APPENDIX C: Source Code Check In Checklist

The Source Code Check In Checklist is a validation checklist that the CM Lead will follow for every email EAI Source Code Check-In Request.

X:\CIO\TechArch\EAI Core Release 3\CM\Code Check-In Checklist.doc

- All sections of the Check-In email request are completed
 - File name
 - Reason for change
 - Detailed description of change
- File location and file contents verified



APPENDIX D: Migration Checklist

The Migration Checklist is a validation checklist that the CM Lead will follow for every email EAI Build Migration Request.

X:\CIO\TechArch\EAI Core Release 3\CM\Migration Checklist.doc

Email

- All sections of the Migration email request are completed
 - Environment
 - Date and time needed
 - Server name
 - Reason for migration

ClearCase

- All source code checked in and commented in ClearCase
- Source code labeled for release
- Code diffed (UNIX “diff” command to display difference between two files) with previous version (where applicable)
- Code diffed (UNIX “diff” command to display difference between two files) with current production version (where applicable)

Release Notes

A release note is a text file containing the bulleted points below:

- Source code version indicated
- Date indicated
- List of files and versions from ClearCase repository indicated (ClearCase “find” command to display baseline within ClearCase.)
- List of changes indicated

Staging

- Source code successfully built without errors



APPENDIX E: MQSI Migration Procedures

X:\CIO\TechArch\EAI Core Release 3\CM\EAI MQSI Production Migration Plan.doc

All MQSeries Integrator (MQSI) configuration items should be managed by a version control system to provide not only a safe storage area for current and previous object versions but also allows detailed tracking and alteration of the configuration items to be stored. MQSI configuration items include message flows, message sets, topics, NEON formats and NEON rules. These items should be configuration managed in ClearCase using the follow naming convention:

- Message sets - \eai\MQSI\dev\messagesets.zip. The zip file will contain a file (called <message set name>.mrp) for each exported message set.
- Message flows - \eai\MQSI\dev\messageflows.zip. The zip file will contain a file (called <message flow name>.xml) for each exported message flow.
- Publish/Subscribe topics - \eai\MQSI\dev\pubsub.top
- NEON formats - \eai\MQSI\dev\<format name>.nnfie
- NEON rules - \eai\MQSI\dev\<application group name>.nnrie

See *Appendix E Export Configuration Items from MQSI* for instructions on exporting configuration items from MQSI.

Export Configuration Items from MQSI

Message Set Export Process

1. In the D:\MQSICM directory on SFANT006, execute EAICMMQSI.bat from the command line. This command executes a java program that will build a batch file. The newly created batch file uses the MQSI mqsimrmimpexp utility to create a separate file for each message set in the MQSI Configuration Manager. The message set files that will be created will have a mrp extension. EAICMMQSI eliminates the need for manually executing mqsimrmimpexp for each individual message set.

Message Flow Export Process

1. Use the NT Control Center GUI to export the message flows from MQSI Configuration Manager.
 - a. Select *File->New Workspace*.
 - b. Select the *Message Flows* tab.
 - c. Highlight the *Message Flows* folder located in the view.
 - d. Select *Message Flow Types->Add to Workspace->Message Flow*.
 - e. Select all the message flows in the list and then select *Finish*.
 - f. Select *File->Export* to export the message flows to your local machine. It is recommended that you use MQSIEport as the filename.
2. Copy the export file to D:\MQSICM directory on SFANT006.
3. Execute EAICMMQSI.bat <export file name> from the command line. This command executes a java program that will build a batch file. The newly created batch file uses the mqsifiltermsgflows functionality that is a part of IC01 supportpac to create a separate file for each of the message flows that has been exported. The message flow files that will be created will have a xml extension. EAICMMQSI eliminates the need for manually exporting each individual message flow or executing mqsifiltermsgflows for each individual message flow.



Publish/Subscribe Topic Export Process

1. Use the NT Control Center GUI to export the publish/subscribe topics from MQSI Configuration Manager.
 - a. Select *File->New Workspace*.
 - b. Select the *Topics* tab.
 - c. Select the *Topic/Users*. If *TopicRoot* is the only entry, then there are no topics to export.
 - d. If there are topics to export, then select the *Message Flows* tab.
 - e. Highlight the *IBMPatterns* folder located in the view.
 - f. Select *Edit->Delete*.
 - g. Select *File->Export* to export the topics to your local machine, specifying pubsub.top as the filename.

NEON Format Export Process

1. On SU35E3 and SU35E14, export the NEON Format using the nnfie command line utility. From /opt/mqi110/bin, execute nnfie -e <format name>.nnfie -m <format name>.
2. Check the nnfie.log file in that directory to ensure that the export was successful.

NEON Rules Export Process

1. On SU35E3 and SU35E14, export the NEON Format using the nnrie command line utility. From /opt/mqi110/bin, execute nnrie -e <application group name>.nnrie -a <application group name>.
2. Check the nnrie.log file in that directory to ensure that the export was successful.

Import Configuration Items to MQSI

Message Set Import Process

1. Use the NT Control Center GUI to delete any existing message sets that will be imported. **NOTE:** If this step is not done, then existing message sets will not be updated.
 - a. Select *File->New Workspace*.
 - b. Select the *Message Sets* tab.
 - c. Select *Message Sets->Add to Workspace->Message Set*.
 - d. From the list, select any existing message set that will be imported and then select *Finish*.
 - e. For each message set that is added to the view, highlight the message set name and then select *Edit->Delete*.
2. Import each message set using the mqsimrmimpexp utility. From the command line utility, execute mqsimrmimpexp -I MQSIMRDB eaiadmin <password> <message set name>.mrp. Message sets will have an .mrp extension.
3. Stop and start the Configuration Manager by executing mqsistop ConfigMgr and mqsistart ConfigMgr, respectively.

Message Flow Import Process

1. Use the NT Control Center GUI to check out any existing message flows that will be imported. Message flows will have an .xml extension. **NOTE:** If this step is not done, then existing message flows will not be updated, unless mqsiimportmsgflows (previously mqsideletemsgflows) utility from IC01 supportpac is used.
 - a. Select *File->New Workspace*.



- b. Select the *Message Flows* tab.
 - c. Highlight the *Message Flows* folder located in the view.
 - d. Select *Message Flow Types->Add to Workspace->Message Flow*.
 - e. From the list, select any existing message flow that will be imported and then select *Finish*.
 - f. For each message flow that is added to the view, highlight the message flow name and then select *Message Flow Types->Check Out*. A symbol (of a key) will appear to the left of the message flow name.
2. Use the NT Control Center GUI to import each message flow.
 - a. Select *File->Import*. Ensure that *Message Flows* is checked, select the location of the xml file and then select *Import*.
 - b. Ensure that the information window that appears does **not** show that “A total of 0 resources were imported into your local repository”. This indicates that the import was not successful and must be redone. If this occurs ensure that you have checked out the message flow.
 3. Use the NT Control Center GUI to check in the newly imported message flows, by selecting *File->Check In->All in Current Workspace*.

Publish/Subscribe Topic Import Process

1. Use the NT Control Center GUI to check out the *TopicRoot* and any existing topics that will be imported.
 - a. Select the *Topics* tab.
 - b. Select the *Topic/Users*.
 - c. Select *TopicRoot* and then select *Topics->Check Out*.
 - d. For each topic that will be imported, select the topic name and then selection *Topics->Check Out*.
2. Use the NT Control Center GUI import function to import the topics. Publish subscribe topics will be in pubsub.top file.
 - a. Select *File->Import*. Ensure that *Topics* is checked, select the pubsub.top file and then select *Import*.
 - b. Ensure that the information window that appears does **not** show that “A total of 0 resources were imported into your local repository”. This indicates that the import was not successful and must be redone. If this occurs ensure that you have checked out the message flow.
3. Use the NT Control Center GUI to check in the newly imported topics, by selecting *File->Check In->All in Current Workspace*

NEON Format Import Process

1. On SU35E3 and SU35E14, import the NEON Format using the nnfie command line utility. From /opt/mqi110/bin, execute nnfie -i <format name>.nnfie -o.
2. Check the nnfie.log file in that directory to ensure that the import was successful.

NEON Rules Import Process

1. On SU35E3 and SU35E14, import the NEON Rules using the nnrie command line utility. From /opt/mqsi110/bin, execute nnrie -i <application group name>.nnrie -o.
2. Check the nnrie.log file in that directory to ensure that the import was successful.



Deployment Process

MQSI Deployment Process

- Message Sets and Message Flows:
 1. Select *File->Deploy->Complete configuration (all types)->Normal.*
 2. Select the *Log* tab.
 3. Select *View->Refresh* until messages from EAIP1 and EAIP2 brokers are received. Ensure that the brokers were able to deploy successfully.
- Topics:
 1. Select the *Topics* tab.
 2. Select *TopicRoot*.
 3. Select *Topics->Deploy->Complete Topics configuration.*
 4. Select the *Log* tab.
 5. Select *View->Refresh* until messages from EAIP1 and EAIP2 broker are received. Ensure that the brokers were able to deploy successfully.
- NEON Formats and Rules:
Run the command line utility mqsinrreload against the affected brokers to re-cache the NEON formats if necessary. On SU35E3, execute mqsinrreload -b EAIP2 and on SU35E14, execute -b EAIP1.

Execution Group Configuration Process

Refer to MQSI User Documentation *MQ Series Integrator 2.0.1 - Using the Control Center*
[ftp://ftp.software.ibm.com/software/ts/mqseries/library/books/bipyar02.pdf](http://ftp.software.ibm.com/software/ts/mqseries/library/books/bipyar02.pdf)

Broker Configuration Process

Refer to MQSI User Documentation *MQ Series Integrator 2.0.1 - Using the Control Center*
[ftp://ftp.software.ibm.com/software/ts/mqseries/library/books/bipyar02.pdf](http://ftp.software.ibm.com/software/ts/mqseries/library/books/bipyar02.pdf)

Message Flow Configuration Process

Refer to MQSI User Documentation *MQ Series Integrator 2.0.1 - Using the Control Center*
[ftp://ftp.software.ibm.com/software/ts/mqseries/library/books/bipyar02.pdf](http://ftp.software.ibm.com/software/ts/mqseries/library/books/bipyar02.pdf)

Configuring User Trace Process

Refer to MQSI User Documentation *MQ Series Integrator 2.0.1 - Using the Control Center*
[ftp://ftp.software.ibm.com/software/ts/mqseries/library/books/bipyar02.pdf](http://ftp.software.ibm.com/software/ts/mqseries/library/books/bipyar02.pdf)

Production Migration Process

1. MQSI configuration must be tested in the staging area brokers on SU35E17. MQSI configuration items will not be migrated to production until this has occurred.



2. Once these items have been validated in the staging area, they will be marked and packaged (in a zip file) for production deployment by EAI. The EAI zip file should contain the following items:
 - Message sets - <message set name>.mrp
 - Message flows - <message flow name>.xml
 - Publish/Subscribe topics - pubsub.top
 - NEON formats - <format name>.nnfie
 - NEON rules - <application group name>.nnrie
3. Once packaged, the zip file will be stored into ClearCase using the follow naming convention:
 - MQSI Production Items - \eai\MQSI\prod\<Application>Interface<mmddyyyy>.zip where mmddyyyy represents the deployment date and not the current date. An example for configuration items related to COD Interfaces is CODInterface03212002.zip.
4. A VDC CM Change request form will be submitted to have these configuration items migrated into production see *Appendix E MQSI Change Request Sample* (for an example). The forms must contain the following information that is relevant to the deployment of MQSI resources:
 - EAI package name
 - MQSI configuration items that will be migrated to production. Include the names of message sets, message flows (including sub-flows) publish/subscribe topics, NEON formats and NEON rules.
 - Execution group configuration (if applicable)
 - Broker configuration (if applicable)
 - Message flow configuration (if applicable)
5. To migrate MQSI configuration items to production, CSC will access the EAI package from ClearCase (in \eai\MQSI\prod directory), unzip the file and perform the necessary steps to import all the MQSI configuration items into the Configuration Manager. (See Section *Importing Configuration Items into MQSI*).
6. Using the NT Control Center GUI, CSC will configure the brokers and execution groups as described in the VDC CM Change request. (See Sections *Execution Group Configuration Process* and *Broker Configuration Process*). These configuration items include:
 - Creating new execution group.
 - Deleting an existing execution group.
 - Adding messages flows to an execution group.
 - Adding message sets to a broker.
7. Deploy changes made to the production environment. Also CSC will configure the deployed message flows as described in the VDC CM Change request. (See Sections *Deployment Processes* and *Production Migration Process*).
 - Starting or stopping a message flow.
 - Modifying the number of instances for a message flow.
 - Setting up user tracing for problem determination.
8. Once the VDC CM Change request has been completed, EAI will validate that these changes through MQSI Operations view. Sufficient validation includes a screen shot of Operations view that is sent by CSC.

Back Out Process

All previous versions of deployed MQSI configuration items will be stored in ClearCase, according to deployment dates. If the situation arises that require the backing out of any configuration item, the previous versions will be available for re-assignment to and re-deployment of the brokers.



MQSI Change Request Sample

CSC Change Request

Please be as detailed as possible

1a. Title of Change:

1b. Detail Description:

MQSI configuration items need to be migrated and deployed into EAI production environment in order to support the transfer of data between COD and FMS and COD and DLOS systems. This configuration has been tested on the SFANT006, SU35E16 and SU35E17 machines. Configuration items that will be migrated are stored in ClearCase under \eai\MQSI\prod\CODInterface03212002.zip. The files in the zip file are COD_FMS_Response.xml, COD_FMS_FinNonfin.xml, FMS_COD_Response.xml, FMS_COD_FinNonfin.xml, COD_FMS_Vendor.xml, COD_FMS_Error_Handler, DLOS_BORROWER_RESPONSE.xml, DLOS_BORROWER_REQUEST.xml, DLOS_ERROR_TRACE1, COD_FMS_V1.mrp, and DLOS_MRM.mrp. These files need to be placed on the NT system, but will be removed prior to the completion of this request. These files will be imported into MQSI on the Ombudsman OAS NT Server and deployed via MQSI to the SU35E3 and SU35E14 Solaris Servers. Validation process will occur during the deployment, by monitoring the system logs (the Event Viewer on Windows NT and the syslog on Solaris) and checking the Operations and Log view of MQSI to make sure the message flows are running.

MQSI Specific Information:

- Import MQSI Message Sets:
 1. COD_FMS_V1
 2. DLOS_MRM
- Import MQSI Message Flows:
 1. COD_FMS_Response
 2. COD_FMS_FinNonfin
 3. FMS_COD_Response
 4. FMS_COD_FinNonfin
 5. COD_FMS_Vendor
 6. DLOS_BORROWER_RESPONSE
 7. DLOS_BORROWER_REQUEST
 8. COD_FMS_Error_Handling
 9. DLOS_ERROR_TRACE1
- Execution Group Configuration:
 1. Create COD_FMS execution group in the EAIP1 and EAIP2 brokers.
 2. Create DLOS execution group in EAIP1 and EAIP2 brokers.
 3. Add COD_FMS_Response, COD_FMS_FinNonfin, FMS_COD_Response, FMS_COD_FinNonfin, and COD_FMS_Vendor to the COD_FMS execution group.
 4. Add DLOS_BORROWER_RESPONSE, and DLOS_BORROWER_REQUEST to the DLOS execution group.
- Broker Configuration:
 1. Add COD_FMS_V1 to EAIP1 and EAIP2 brokers.
 2. Add DLOS_MRM to EAIP1 and EAIP2 brokers.
- Message flow configuration



1. Give COD_FMS_FinNonfin 2 additional instances.
2. Stop COD_FMS_Vendor.
3. Start User Trace for COD_FMS_FinNonfin.

2. Systems affected: EAI Production NT (Ombudsman-OAS)



APPENDIX F: EAI Build Package & Migration Procedures

The EAI Build Package and Migration Procedures are detailed step-by-step instructions on how EAI Builds are created, packaged and migrated.

Code Migration for Development and Test UNIX Servers

1. Log onto su35e16 as "mqm"

ClearCase repository is located on su35e16

2. Execute the ClearCase profile

```
. /cc/vobstore/admin/scripts/ccenvlist.sh
```

3. Set the "mqm" ClearCase view for the specified release of code

Example for latest release of code

```
cleartool setview mqm_def_view
```

Example for release 1 SIT version of code

```
cleartool setview mqm_R1.0SIT_view
```

4. Recursively copy the /vobs/eai/eaiCOD directory to the ~mqm/tmp directory

```
cd ~mqm/tmp  
cp -r /vobs/eai/eaiCOD ./
```

Check available disk space using the "df -k" command at the UNIX prompt

5. Tar the ~mqm/tmp/eaiCOD directory

```
cd ~mqm/tmp  
tar -cvf eaiCOD.tar eaiCOD
```

6. FTP the eaiCOD.tar tar file to the target UNIX server

7. Log onto the target UNIX server as "mqm"

Check available disk space using the "df -k" command at the UNIX prompt

8. Extract the eaiCOD.tar tar file in the "mqm" home directory of the target UNIX server

```
cd ~mqm  
tar -xvf eaiCOD.tar
```

9. Rename the ~mqm/eaiCOD directory to ~mqm/eaicodr1

```
cd ~mqm  
mv eaiCOD eaicodr1
```

10. Update the "mqm" .profile file to include the appropriate environment variables

```
EAIDIR=~mqm/eaicodr1  
. $EAIDIR/scripts/ftfenvlist.sh  
. $EAIDIR/scripts/amienvlist.sh
```



Only update this if the variables are not already set in the "mqm" .profile file

11. Execute the "mqm" .profile file

• `~mqm/.profile`

12. Use the Makefile to execute the build

```
cd $EAIDIR  
make debug
```

- Scroll through the terminal to check for errors.
- The Makefile needs to be modified when new source code is added to the build. This should be coordinated between the developer and person running the build.
- Some common errors occur because of code compilation, path and classpath not being set appropriately, and permissions.

13. Update the following site specific files

- \$EAIDIR/config/EAICONF.LOG
 - Appropriate path for LOGGING_PATH_NAME
LOGGING_PATH_NAME /export/home/mqm/eaicodr1/logfiles
- \$EAIDIR/scripts/distartstop.sh
 - Appropriate Data Integrator Node name
DINODES="Data_Integrator_Node";
- \$EAIDIR/scripts/dclientstartstop.sh
 - Appropriate Queue Manager and Data Integrator Node name
DINODES="Queue_Manager";
DINODES="Data_Integrator_Node";
- \$EAIDIR/scripts/mqsistartstop.sh
 - Appropriate MQSI Broker name
BROKERS="Queue_Manager";
- \$EAIDIR/scripts/mqstartstop.sh
 - Appropriate Queue Manager name
QMGRS="Queue_Manager";
- \$EAIDIR/scripts/ftfenvlist.sh
 - Appropriate Local Queue Manager name
LQM=Queue_Manager; export LQM
- \$EAIDIR/scripts/profile.txt
 - Appropriate path for EAIDIR
EAIDIR=/export/home/mqm/eaicodr1; export EAIDIR



Code Migration to Production UNIX Servers

Production Code Migration procedures are detailed step-by-step instructions on how EAI Builds are delivered to production.

1. Log onto su35e16 as "mqm"
ClearCase repository is located on su35e16

2. Execute the ClearCase profile
 - . /cc/vobstore/admin/scripts/ccenvlist.sh

3. Set the "mqm" ClearCase view for the specified release of code
Example for latest release of code

```
cleartool setview mqm_def_view
```

Example for release 1 SIT version of code

```
cleartool setview mqm_R1.0SIT_view
```

4. Recursively copy the /vobs/eai/eaiCOD directory to the ~mqm/tmp directory

```
cd ~mqm/tmp  
cp -r /vobs/eai/eaiCOD ./
```

Check available disk space using the "df -k" command at the UNIX prompt

5. Tar the ~mqm/tmp/eaiCOD directory

```
cd ~mqm/tmp  
tar -cvf eaiCOD.tar eaiCOD
```

6. FTP the eaiCOD.tar tar file to a migration staging area

7. Log onto the target UNIX server as "mqm"

Check available disk space using the "df -k" command at the UNIX prompt

8. Extract the eaiCOD.tar tar file in the "mqm" home directory of the target UNIX server

```
cd ~mqm  
tar -xvf eaiCOD.tar
```

9. Rename the ~mqm/eaiCOD directory to ~mqm/eaicodr1

```
cd ~mqm  
mv eaiCOD eaicodr1
```

10. Update the "mqm" .profile file to include the appropriate environment variables

```
EAIDIR=~mqm/eaicodr1  
. $EAIDIR/scripts/ftfenvlist.sh  
. $EAIDIR/scripts/amienvlist.sh
```

Only update this if the variables are not already set in the "mqm" .profile file



-
11. Execute the "mqm" .profile file
`. ~mqm/.profile`
 12. Use the Makefile to create the eaibuild.tar tar file for Production
`cd $EAIDIR
make archive`
 13. Migrate the to the IST environment. FTP the eaibuild.tar to the IST environment
 14. Validate to ensure the build is ready for Production. After successful execution of the migration validation procedures (*Appendix G Migration Validation Procedures*), the CM lead will submit the change request to migrate the new EAI build to Production.
 15. Submit a change request using the ClearQuest ECM tool including instructions for migrating the build (eaibuild.tar).



APPENDIX G: Migration Validation Procedures

Migration Validation Procedures are detailed step-by-step instructions on how to validate EAI Builds after they have been installed on UNIX servers.

Use "mqm" to execute the diclientstartstop.sh, dirmonstart.sh, distartstop.sh, and mqstartstop.sh
Use "db2inst1" to execute the mqsistartstop.sh

1. Execute diclientstartstop.sh script to start Data Integrator client

Check Data Integrator client processes are not running
`ps -ef | grep ftf`

Start Data Integrator client
. \$EAIDIR/diclientstartstop.sh start

Check Data Integrator processes have started
`ps -ef | grep ftf`

```
ftfsdr -lqm $QMGRS -nodename $DINODES -cfile $EAIDIR/config/ftfconfig.ini
ftfrcv -lqm $QMGRS -nodename $DINODES -cfile $EAIDIR/config/ftfconfig.ini
ftfmgr -lqm $QMGRS -nodename $DINODES -cfile $EAIDIR/config/ftfconfig.ini
```

Log files located at \$EAIDIR/logfiles/ftf\$DINODES.out

2. Execute diclientstartstop.sh script to stop Data Integrator client

Check Data Integrator processes are running
`ps -ef | grep ftf`

```
ftfsdr -lqm $QMGRS -nodename $DINODES -cfile $EAIDIR/config/ftfconfig.ini
ftfrcv -lqm $QMGRS -nodename $DINODES -cfile $EAIDIR/config/ftfconfig.ini
ftfmgr -lqm $QMGRS -nodename $DINODES -cfile $EAIDIR/config/ftfconfig.ini
```

Stop Data Integrator client
. \$EAIDIR/diclientstartstop.sh stop

Check Data Integrator processes have ended
`ps -ef | grep ftf`

Log files located at \$EAIDIR/logfiles/ftf\$DINODES.out

3. Execute dirmonstart.sh script to start Directory Monitoring

Check Directory Monitoring processes are not running
`ps -ef | grep dirmon`

Start Directory Monitoring
. \$EAIDIR/dirmonstartstop.sh start

Check Directory Monitoring processes have started



```
ps -ef | grep dirmon  
  
dirmon $EAIDIR/config/dirmon.ini
```

Log file located at /home/mqm/eaicodr1/logfiles/dirmon.log

4. Execute distartstop.sh script to start Data Integrator
Check Data Integrator processes are not running

```
ps -ef | grep ftf
```

Start Data Integrator
. \$EAIDIR/distartstop.sh start

Check Data Integrator processes have started

```
ps -ef | grep ftf
```

```
ftfsdr -lqm $qmgr -cfile $EAIDIR/config/ftfconfig.ini  
ftfrcv -lqm $qmgr -cfile $EAIDIR/config/ftfconfig.ini  
ftfmgr -lqm $qmgr -cfile $EAIDIR/config/ftfconfig.ini
```

Log files located at \$EAIDIR/logfiles/ftf\$DINODES.out

5. Execute distartstop.sh script to stop Data Integrator
Check Data Integrator processes are running

```
ps -ef | grep ftf
```

```
ftfsdr -lqm $qmgr -cfile $EAIDIR/config/ftfconfig.ini  
ftfrcv -lqm $qmgr -cfile $EAIDIR/config/ftfconfig.ini  
ftfmgr -lqm $qmgr -cfile $EAIDIR/config/ftfconfig.ini
```

Stop Data Ingegrator
. \$EAIDIR/distartstop.sh stop

Check Data Integrator processes have ended

```
ps -ef | grep ftf
```

log files located at \$EAIDIR/logfiles/ftf\$DINODES.out

6. Execute mqsistartstop.sh script to start MQSI
Check MQSI processes are not running

```
ps -ef | grep db2inst1
```

Start MQSI
. \$EAIDIR/mqsistartstop.sh start

Check MQSI processes have started

```
ps -ef | grep db2inst1
```

```
bipservice $QMGR  
bipbroker $QMGR
```



DataFlowEngine \$QMGR

7. Execute mqsistartstop.sh script to stop MQSI

Check MQSI processes are running

```
ps -ef | grep db2inst1
```

```
bipservice $QMGR  
bipbroker $QMGR  
DataFlowEngine $QMGR
```

Stop MQSI

```
. $EAIDIR/mqsistartstop.sh stop
```

Check MQSI processes have ended

```
ps -ef | grep
```

8. Execute mqstartstop.sh script to start MQ Series

Check MQ Series processes are not running

```
ps -ef | grep amq
```

Start MQ Series

```
. $EAIDIR/mqstartstop.sh start
```

Check MQ Series processes have started

```
ps -ef | grep amq
```

```
/opt/mqm/bin/amqzfuma -m $QMGRS  
amqzxma0 -m $QMGRS  
amqhasmx $QMGRS /var/mqm  
/opt/mqm/bin/amqrrmfa -t2332800 -s2592000 -p2592000 -g518400  
amqzlaa0 -m$QMGRS -fip0  
amqpcsea $QMGRS ?
```

Log files located at EAIDIR/logfiles/mq\$QMGRS.out

9. Execute mqstartstop.sh script to stop MQ Series

Check MQ Series processes are running

```
ps -ef | grep amq
```

```
/opt/mqm/bin/amqzfuma -m $QMGRS  
amqzxma0 -m $QMGRS  
amqhasmx $QMGRS /var/mqm  
/opt/mqm/bin/amqrrmfa -t2332800 -s2592000 -p2592000 -g518400  
amqzlaa0 -m$QMGRS -fip0  
amqpcsea $QMGRS ?
```

Stop MQ Series

```
. $EAIDIR/mqstartstop.sh stop
```

Check MQ Series processes have ended



```
ps -ef | grep amq
```

Log files located at EAIDIR/logfiles/mq\$QMGRS.out